第6章　ブラックモデル

Fisher Black(ブラック博士) は 1976 年に”The Pricing of Commodity Contracts”, Journal of Financial Economics を発表し、先物オプションを評価できるように 1973 年発表の無配当株式用のブラック ショールズ モデル (以下 **BS モデル**) を拡張した。そのモデルは**ブラックモデル** (または**ブラック 76** モデル) と呼ばれ、債券関連のヨーロピアンタイプのオプションを評価する標準モデルとなっている。

このブラックモデルを 2 項ツリーやモンテカルロシミュレーションで表現することで、アメリカンタイプのオプションが評価できる。

6.1　ブラック価格式と債券先物オプション計算例

多くの読者[*1] は BS モデル式やそのオプション価格式を一度は見たことがあると思われるが、債券市場で標準的に使用されているオプション計算は正確にはブラックモデルを使用した価格式である。

BS モデルはスポット価格 S_t の収益率が正規分布に従うと仮定する。モデル式は r をオプション満期時点のリスクフリーレート、σ をボラティリティとし、次式で表される。

$$\frac{dS_t}{S_t} = rdt + \sigma dW_t^Q \tag{6.1}$$

ここで時点 t, 満期 T の割引債価格を $P_{(t,T)}$ で表すと、フォワード価格 (フォワードの満期は T) は

$$F_t := \frac{S_t}{P_{(t,T)}} \tag{6.2}$$

となるが[*2] 、ブラックモデルはこのフォワード価格 F_t の収益率が正規分布すると仮定している点が異なり、モデル式は次式である。

$$\frac{dF_t}{F_t} = \sigma dW_t^T \tag{6.3}$$

[*1] この章の読者はデリバティブの初歩レベルを理解していることを想定している。自信の無い読者はデリバティブの標準的なテキスト Hull[34] を辞書代わりに参照して頂きたい。

[*2] 例えば、満期 T の連続複利を R とすると、$P_{(t,T)} = e^{-R(T-t)}$ であり、フォワード価格は $F_t = S_t/P_{(t,T)} = S_t \times e^{R(T-t)}$。

フォワード価格式も含め、これらの式が何を意味しているかはこの章を通じて説明して行くので、ここでは眺めておくだけで良い。(2 つのモデルの違いは 6.5.2 節を参照)

オプションのモデルを理解する場合、確率変動するものが何で、その確率分布が何かを意識する必要がある。例えばブラックモデルではフォワード価格が対数正規分布に従うと仮定する。第 7 章のノーマルモデルはフォワードレート (金利) が正規分布することを仮定し、第 9 章のハルホワイトモデルはイールドカーブに現れないショートレートが正規分布するとしている。

これらは一般的な注意点であるが、モデル式からオプション価格式を導出する説明は非常に長いものとなるので、この章では初めに計算例を示し、その後でモデルを理解する為のイメージ的な補足を行う。尚、所々で結論を先に述べるトップダウン的な説明を行っている旨、了承頂きたい。

6.1.1　ブラック価格式の数値例

図 6.1 は CBOT (Chigago Board of Trade) に上場されていた米国 5 年債先物 2023 年 9 月限のコールオプションの評価シートで

B2 セル:	コール = 1 (プット=−1)	
B3 セル:	取引日：	2023 年 7 月 21 日
B4 セル:	オプション満期日：	2023 年 8 月 26 日 (36 日)
B5 セル:	先物価格：	107.21094 ドル (32 進で 107-6 3/4)
B6 セル:	行使価格：	107.50 ドル
B7 セル:	ボラティリティ：	5.20%
B8 セル:	リスクフリーレート：	5.0% (連続複利)

等が与えられ、オプション価格 (NPV) 0.5616 ドル (B21 セル) が算出されている。

この図では B21 セルと F2 セルの 2 箇所でオプション価格 0.5616 ドルが算出されている。B21 セルは次の式 (6.4) から (6.6) を使って算出した値であり、F 列は QuantLib の関数による計算となる。

初めに B21 セルの計算を見ていこう。ブラックモデルのオプション価格式

	A	B	C	D	E	F	G
1	債券先物 (Blackモデル)		(B列 数式/コメント)		QuantLib		(F列 数式/コメント)
2	pc: コール=1, プット= -1	1			NPV	0.5616	=bsOption(B2,B3,B4,B5,B6,B7,B8, 0, E2)
3	取引日	2023/7/21			delta	0.4356	=bsOption(B2,B3,B4,B5,B6,B7,B8, 0, E3)
4	満期日	2023/8/26			gamma	0.2240	=bsOption(B2,B3,B4,B5,B6,B7,B8, 0, E4)
5	F: 先物価格	107.21094	=107-6 3/4		vega	13.2032	=bsOption(B2,B3,B4,B5,B6,B7,B8, 0, E5)
6	X : 行使価格	107.50000			theta	-0.0095	=bsOption(B2,B3,B4,B5,B6,B7,B8, 0, E6)
7	σ: ボラティリティー	5.2000%					
8	r : リスクフリーレート	5.0000%	連続複利		NPVからvol算出		
9	(基本的計算項目)				0.7	6.2456%	=bsOption(B2,B3,B4,B5,B6,B7,B8,E9, "vol")
10	満期日数	36	=B4-B3				
11	T : オプション満期年数	0.09863	=B10/365				
12	D_T : ディスカウント	0.99508	=EXP(-B8*B11)				
13	σ√T	0.01633	=B7*B11^0.5				
14	(オプション計算項目)						
15	d1*pc	-0.1567	=B2*(LN(B5/B6)/B13+B13/2)				
16	d2*pc	-0.1730	=B2*(LN(B5/B6)/B13-B13/2)				
17	N(d1*pc)	0.4377	=NORMDIST(B15,0,1,1)				
18	N(d2*pc)	0.4313	=NORMDIST(B16,0,1,1)				
19	n(d1*pc)	0.3941	=NORMDIST(B15,0,1,0)				
20	ブラックコア	0.5644	=B2*(B5*B17-B6*B18)				
21	オプション NPV	0.5616	=B12*B20				
22	デルタ	0.4356	=B2*B12*B17				
23	ガンマ	0.2240	=B12*B19/(B5*B13)	セータ第1項		3.4977	=B5*B19*B7/(2*B11^0.5)
24	ベガ	13.2032	=B12*B5*B11^0.5*B19	セータ第2項		-2.3465	=B2*-B8*B5*B17
25	セータ	-0.0095	=-B12*SUM(F23:F25)/365	セータ第3項		2.3183	=B2*B8*B6*B18

図 6.1: 5 年債先物 9 限月コールオプションの評価

(以下**ブラック価格式**) は c をコール価格、p をプット価格として、次式となる。

$$\underset{(\text{コール})}{c} = \underbrace{D_T}_{\substack{\text{ディスカウント}\\\text{ファクター}}} \times \Big[\underbrace{\underbrace{F}_{\substack{\text{先物}\\\text{価格}}} \overbrace{\underbrace{N(d_1)}_{\substack{\text{満期時}\\\text{デルタ}}}}^{\substack{F\text{ 測度で}\\\text{行使価格を}\\\text{超える確率}}} - \underbrace{X}_{\substack{\text{行使}\\\text{価格}}} \overbrace{N(d_2)}^{\substack{\text{行使価格を}\\\text{超える確率}}}}_{\text{ブラック コア}} \Big] \tag{6.4}$$

$$\underset{(\text{プット})}{p} = -D_T \times \big[\, FN(-d_1) - XN(-d_2) \,\big] \tag{6.5}$$

$$d_1 = \frac{ln(F/X)}{\sigma\sqrt{T}} + \frac{\sigma\sqrt{T}}{2} \;,\quad d_2 = \frac{ln(F/X)}{\sigma\sqrt{T}} - \frac{\sigma\sqrt{T}}{2} \tag{6.6}$$

F : 先物価格やフォワード金利等　　X : 行使価格

σ : ブラックのボラティリティ　　T : 満期年

D_T : オプション満期日のディスカウントファクター等

$N(\cdot)$: 標準正規分布の累積密度関数

この計算式に図 6.1 の数値を当てはめ、それぞれの値の意味を説明しよう。ただ 意味不明な箇所は不明なまま、先に進んで欲しい (数式に記したルビも同様)。この章を読み終えて、再度ここの説明を読めば、理解できると思われる。

B2〜B8 : 計算の前提となる 7 つの入力項目。

- B2 セルの “ pc ” は putcall 用フラッグ。(A15 セル以下で現れる “ ***pc** ” は 1 または -1 を掛けることを示す)
- B8 セルのリスクフリーレートはオプション満期までのレートを設定。

◇ ここでは計算を簡便にする為、連続複利レート 5%を設定。

◇ 債券先物オプションの場合、国債のレポレートを使用する場合が多い。

B11: オプション満期年数： 満期までの年数はカレンダーベースな為、$\underset{(B10\ セル)}{36\ 日}$ ÷365 で計算。(通常、日数計算は Act/365)

B12: D_T ディスカウント： 価格式 (6.4) の右辺先頭にある D_T は T 年満期のディスカウントファクター。

- D_T の計算は評価する商品と市場慣行で異なる点に注意。
- B8 セルを連続複利レートとした為、B12 セルでは連続複利式によりディスカウントファクターを算出。
- レポレートなら、単利でディスカウントファクターを算出。

B16: d2*pc： 式 (6.6) より d_2 は次式で算出。(B2 セルで pc=1 とした為、以下の数式や説明では pc 部分は省略)

$$\begin{aligned} d_2 &= \frac{\ln\Big(\overset{F:\ 先物価格}{107.21\ ドル}\Big/\overset{X:\ 行使価格}{107.50\ ドル}\Big)}{5.20\% \times \sqrt{0.0986\ 年}} - \frac{\overset{\sigma}{5.20\%} \times \sqrt{\overset{T:\ 満期年}{0.0986\ 年}}}{2} \\ &= \underset{(B16\ セル)}{-0.1730} \end{aligned} \tag{6.7}$$

- d_2 の意味は " 行使価格 107.50 ドルは標準正規分布上、中心のゼロより若干右側 0.1730 に換算されている "。(この説明は 6.8.2 節参照)
- " 中心 " は先物価格 107.21 がゼロとなっている。(正規分布でイメージすることが重要)

B18: N(d2*pc)： $N(d_2)$ はエクセル関数の $= NormDist($値$,$ 平均 $= 0,$ 標準偏差 $= 1, True = 1)$ で算出され、0.4313。

- この値の意味は " 満期時点に債券先物価格が 107.50 ドルを超える確率が 43.13% "。(正規分布でイメージすること)
- 43.13% の " 確率 " は**リスク中立確率**。(6.5 節で説明)

B17: N(d1*pc)、 B22: デルタ： 式 (6.6) より d_1 を算出。

$$d_1 = \underset{d_2}{-0.1730} + \underset{\sigma}{5.20\%} \times \underset{T:\ 満期年}{\sqrt{0.0986\ 年}} = -0.1567$$

- $N(d_1) = 0.4377$ は $N(d_2)$ と同様 満期時点で債券先物価格が 107.50 ドルを超える確率。
- この確率は**フォワード中立確率**。(式 6.4 のルビでは " F 測度 " と表記。)

- $N(d_2)$ のリスク中立確率と $N(d_1)$ のフォワード中立確率の違いは 6.5 節で説明。

  (ここではこの 2 つを区別する必要はなく、$N(d_1)$, $N(d_2)$ は**インザマネー**[*3] となる確率 (43%程度) が計算されていることを理解することが重要。)
- 尚 $N(d_1) = 0.4377$ はオプション**満期時点のデルタ**。
  - ◇ デルタの計算式は式 (6.8) : $D_T \times N(d_1)$。
  - ◇ " 満期時に 43%の確率でインザマネーになる " とは " 先物を約 0.43 枚保有 " に同じ。

B20: ブラックコア : 式 (6.4) で " ブラックコア " というルビ部分を計算。

- " $FN(d_1) - XN(d_2)$ " はキャップやスワップションの計算でも同じ数式の為、" コア " と呼ぶ。

$$\underbrace{\overset{(B20)}{0.5644\text{ ドル}}}_{\text{ブラックコア}} = \underbrace{\overset{(B5)}{107.21\text{ ドル}}}_{F:\text{ 先物価格}} \times N(\underbrace{\overset{(B17)}{0.4377}}_{d1}) - \underbrace{\overset{(B6)}{107.50\text{ ドル}}}_{X:\text{ 行使価格}} \times N(\underbrace{\overset{(B18)}{0.4313}}_{d2})$$

(ブラックコアの単位) 上式右辺では先物価格や行使価格に確率を掛けているだけなので、算出される左辺の値の単位は先物価格や行使価格と同じ単位。(ブラックコアの単位は重要)

B21: オプション NPV : 式 (6.4) よりオプション価格を算出。[*4]

$$\underbrace{\overset{(B13)}{0.5616\text{ ドル}}}_{\text{オプション }NPV} = \underbrace{\overset{(B12)}{0.99508}}_{\substack{D_T:\\ \text{ディスカウント}}} \times \underbrace{\overset{(B20)}{0.5644\text{ ドル}}}_{\text{ブラックコア}}$$

(オプション NPV の単位) 右辺の計算はブラックコアにディスカウントファクターを掛けているだけなので、算出される NPV はブラックコアと同じ単位。(この例では先物価格と同じ単位)

この第 6 章はこれらの計算の背後にある理論の理解と算出された数値の解釈ができるようになることが目的である。

6.1.2 ブラックモデルのグリークス計算式

次にブラックのグリークス計算式をまとめておく。グリークスはオプションのリスク管理で必要不可欠なデータであり、計算例は図 6.1 の A22〜B25

[*3] ペイオフ額 (式 6.19 参照) がプラスのオプションの状態を**イン ザ マネー** (In the money, ITM)。ペイオフ額がゼロの状態を**アット ザ マネー** (**At the money, ATM**)、マイナスの状態を**アウト オブ ザ マネー** (**Out of the money, OTM**) と呼ぶ。

[*4] オプション価格は**オプションプレミアム**とも呼ばれるが、QuantLIB の NPV メソッドに合わせて NPV(Net Present Value) と呼んだ。

セルを参照。同図 E2〜F6 セルに QuantLib のメソッドで同様のグリークスを算出させた。(この部分の説明は 6.3.3 節を参照)

$$\underset{(call)}{\text{デルタ}}:\frac{\partial c}{\partial F} = D_T N(d_1)\ ,\quad put\ :\ -D_T N(-d_1) \tag{6.8}$$

$$\text{ガンマ}:\frac{\partial^2 c}{\partial F^2} = D_T \frac{n(d_1)}{F\sigma\sqrt{T}} \tag{6.9}$$

$$(\text{標準正規分布 密度関数}:n(d) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}d^2}\ )$$

$$\text{ベガ}:\frac{\partial c}{\partial \sigma} = D_T F\sqrt{T}n(d_1) \tag{6.10}$$

$$\underset{(call)}{\text{セータ}}:\frac{\partial c}{\partial T} = -D_T\Big[\frac{Fn(d_1)\sigma}{2\sqrt{T}} - rFN(d_1) + rXN(d_2)\Big] \tag{6.11}$$

$$put\ :\ -D_T\Big[\frac{Fn(d_1)\sigma}{2\sqrt{T}} + rFN(-d_1) - rXN(-d_2)\Big]$$

- 式 (6.11) のセータは年単位で算出。1 日単位のセータは 365 日で割る点に注意。

尚、ブラックモデルのグリークスの使い方は省略したが、ノーマルモデルでは 7.5.1 節で数値例を挙げで説明した。2 つのモデルのグリークスの使い方は同じとなる為。

6.2　収益率の正規分布とブラック価格式のイメージ

ここでは 6.1 節で使用したオプション価格式を入門レベルで説明し、計算式の各項目が何を算出しているかを確認する。

6.2.1　収益率の分布

まず ブラックモデルが仮定する " 証券価格の収益率が正規分布する " 点について確認しよう。

図 6.2 は 5 年債先物の清算価格 (B 列 lastPRC) のヒストリーデータであり、C 列 smpRTN (simple return の略) は前日からの収益率が計算されている。

例えば、2023 年 7 月 21 日の 0.022%は 7 月 20 日の価格 107.2343750 と 21 日の 107.2578125 から、次式で算出。

$$\frac{\overset{(2023/07/21)}{107.2578125} - \overset{(2023/07/20)}{107.2343750}}{107.2343750} = \overset{(C2\ \text{セル})}{0.022\%} \tag{6.12}$$

計算式 C2=(B2-B3)/B3　D2=LN(B2/B3)　E2=STDEV(D2:D5)　F2=E2*250^0.5

	A	B	C	D	E	F
1	tradeDT	lastPRC	smpRTN	logRTN	StDev5ds	HistVol5ds
2	2023/07/21	107.2578125	0.022%	0.022%	0.2855%	4.514%
3	2023/07/20	107.2343750	-0.522%	-0.523%	0.3008%	4.756%
4	2023/07/19	107.7968750	0.102%	0.102%	0.2467%	3.900%
5	2023/07/18	107.6875000	0.000%	0.000%	0.3993%	6.313%
6	2023/07/17	107.6875000	0.109%	0.109%	0.4980%	7.874%

HistPrc

図 6.2: 5 年債先物の日次収益率 (価格 lastPRC は CME の清算値段)

オプション関連の場合、この収益率は D 列の計算のように自然対数 ln を使用し、**連続複利の収益率**[*5] として次のように計算する。

$$\text{連続複利 収益率} = \ln\left(\frac{\text{時点}_2\text{の価格}}{\text{時点}_1\text{の価格}}\right) \tag{6.13}$$

$$= \ln\frac{107.2578125}{107.2343750} = \overset{(D2\text{ セル})}{0.022\%} \tag{6.14}$$

図 6.3 は 2023 年 9 月限の 2022 年 1 月 3 日から 2023 年 7 月 21 日までの 139 営業日の前日比の収益率の度数分布と統計データである。

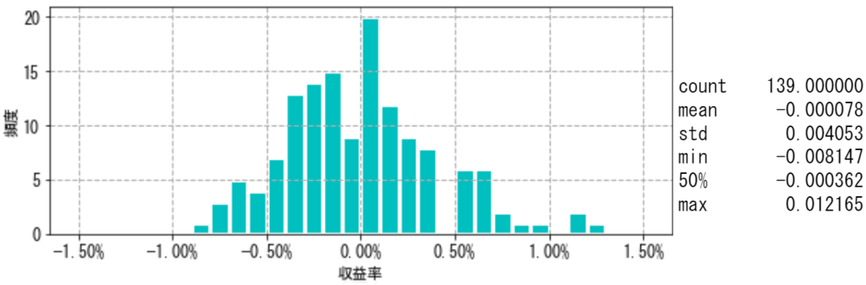


図 6.3: 5 年債先物 日次収益率の分布

このグラフのコードは??節で説明する。ここではこの棒グラフの次の点を確認しよう。

- 棒グラフの X 軸は収益率, Y 軸は発生頻度。
- X 軸 0.00%〜0.05%区間が 20 回で最も多く出現。
- この期間の収益率の**平均値** (**μ: ミューまたは mean**) は-0.0078%。
- **標準偏差** (**σ: シグマまたは standard deviation**) は 0.4053%。

[*5] (**指数、対数の経済的解釈**)

- r を収益率として $e^r \simeq 1 + r$ (e^x の 1 次までの**テーラー展開**) なので、e^r は元本 1 と収益 r の合計額を計算。
- 一方 $\ln(e^r) = r$ から、対数 ln は $1 + r$ から収益率 r のみを取り出す関数。
- 式 (6.14) は価格比で“$1 + r$”を計算させ、収益率を取り出している。

この棒グラフの形から、“ 先物価格の収益率は正規分布 ” の仮定は正しいように思われる。**正規分布の記号**として、**N(平均, 分散)** を使えば、図 6.3 は次のように書く。[*6]

$$\frac{dF_t}{F_t} \quad \sim \quad N(\underbrace{0\%}_{\text{平均}}, \underbrace{0.4053\%^2}_{\sigma^2:\text{ 分散}}) \tag{6.15}$$

左辺の分数は時点 t の先物価格 F_t の収益率を式 (6.13) を使い、次のように定義している。

$$\frac{dF_t}{F_t} \quad := \quad \lim_{\delta \to 0} \frac{F_{t+\delta} - F_t}{F_t} \quad = \quad \ln \frac{F_{t+\delta}}{F_t} \tag{6.16}$$

尚、式 (6.15) を確率過程の方程式として書き直した式が式 (6.3) である。(正規分布と確率過程の方程式の関係は 6.8 節で説明)

6.2.2 ヒストリカル vs インプライド ボラティリティ

オプションの世界では標準偏差を年率ベースに直した値は**ボラティリティ**と呼ばれ、特に式 (6.16) で計算される収益率のボラティリティは **Black volatility**(**ブラックのボラティリティ**) として参照される。

図 6.3 の標準偏差 0.4053%は 1 日当たりで算出されている。1 年を 250 営業日とすると、1 年間の収益率の標準偏差はいわゆる**ルート T 倍法**[*7] により、

$$\overbrace{0.4053\%}^{\substack{\text{一日当たり}\\\text{標準偏差}}} \times \overbrace{\sqrt{250\text{ 日}}}^{\text{ルート } T \text{ 倍法}} \quad = \quad \overbrace{6.408\%}^{\substack{\text{年率換算}\\\text{標準偏差}}} \tag{6.17}$$

と計算され、ボラティリティは6.408%となる。このように**原資産** (underlying asset) の価格データを基に計算されたボラティリティを**ヒストリカル ボラティリティ**と言う。

一方 図 6.1 では B7 セルでボラティリティ5.20%を与えて オプション価格を計算したが、このボラティリティは**インプライド ボラティリティ**と呼ばれる。

インプライド ボラティリティとはブラックのオプション価格式 (6.4), (6.5) を使用し、市場で売買されているオプション価格から、逆算したボラティリティである。

インプライド 5.20%とヒストリカル 6.408%の 2 つのボラティリティを較べると比較的似た値となっていることが判る。市場で建値されたオプション価格の妥当性を確認するにはヒストリカル ボラティリティと較べることが重要である。

[*6] ブラックのモデルでは平均値をゼロにするテクニックが使用される。6.5.2 節参照。

[*7] X_1 と X_2 を 1 日目と 2 日目の証券の収益率を表す確率変数とし、$X_1 \sim N(\mu, \sigma^2)$、$X_2 \sim N(\mu, \sigma^2)$ のように同じ正規分布に従い、かつ、X_1 と X_2 が独立の場合、$X_1 + X_2 \sim N(\mu + \mu, \sigma^2 + \sigma^2) = N(2\mu, 2\sigma^2)$ が成り立つ。従って 2 日間の標準偏差は $\sqrt{2}\,\sigma$。

(ヒストリカルボラティリティの日数)

図 6.3 で算出されたヒストリカル ボラティリティは 2022 年 12 月 30 日から 2023 年 7 月 21 日までの 140 日間のデータを使用して算出した。価格推移を示した図 6.2 の F 列 (HistVol5ds) は 5 日間のヒストリカル ボラティリティを各取引日毎に算出させた。

140 日間のヒストリカル ボラティリティ6.408%よりも 7 月 21 日の 5 日間ヒストリカルボラティリティ4.514%のほうがインプライド ボラティリティ5.20%には近い値となっている。この理由として、インプライド ボラティリティは原資産の直近の値動きに影響される為であろう。

ヒストリカル ボラティリティの計算式は図 6.2 の上部ボックス内に記したが、以下簡単に説明しておく。

日次標準偏差： E 列ではエクセルの関数 $= StDev(\cdot)$ で、日次収益率の標準偏差を算出。

$$\underbrace{\overset{(E2\text{ セル})}{0.2855\%}}_{\text{日次標準偏差}} = StDev\big(\underbrace{D2 : D5}_{\text{日次収益率}}\big)$$

ルート T 倍法： F 列では E 列の数値をルート T 倍法により年率化し、5 日間ヒストリカル ボラティリティを算出。

$$\underbrace{\overset{(F2\text{ セル})}{4.514\%}}_{\substack{\text{5 日間ヒストリカル}\\ \text{ボラティリティ}}} = \overset{(E2\text{ セル})}{0.2855\%}\underbrace{\times \sqrt{250\text{ 日}}}_{\text{ルート } T \text{ 倍法}}$$

6.2.3 先物価格の分布の確率

正規分布は平均と標準偏差の 2 つのパラメータで分布の形状が決定される。収益率が正規分布することを前提としたブラックモデルでは平均をゼロ、標準偏差をインプライド ボラティリティとすることで、先物の収益率の分布を確定させている。分布が決まれば、将来の先物価格は確率を伴って特定することができる。

例えば、図 6.1 の B13 セル 1.633%$(= 5.2\% \times \sqrt{36/365})$ はオプション満期日 (36 日後) の標準偏差をインプライド ボラティリティ (5.2000%) からルートＴ倍法で算出した値であり、5 年債先物の収益率が 1 標準偏差の範囲に収まる確率は約 68%となる。[*8] このことを確率を表す記号 $\boldsymbol{Pr\{\cdot\}}$ を使えば、次式のよ

[*8] この確率は正規分布より計算。正規分布の特性は統計学の教科書を参照。

うに表せる。

$$Pr\{\underset{(-1\sigma)}{-1.633\%} \le \overset{\text{5 年債先物}}{\text{収益率}} \le \underset{(+1\sigma)}{1.633\%}\} = \text{約 } 68\%$$

この式に収益率の定義式 (6.16) と“伊藤の公式”を使用して、オプション満期日の 5 年債先物価格が次のように算出できる。

$$\underset{(-1\sigma)}{-1.633\%} + \overbrace{\underset{(\frac{1}{2}\sigma^2 T)}{0.0133\%}}^{\substack{\text{伊藤の公式}\\\text{からの修正項}}} \le \underset{\substack{\text{満期時価格}\\\text{(8 月 26 日)}}}{ln(F_T)} - \underset{\text{7 月 21 日価格}}{ln(107.2109)} \le \underset{(+1\sigma)}{1.633\%} + \overbrace{\underset{(\frac{1}{2}\sigma^2 T)}{0.0133\%}}^{\substack{\text{伊藤の公式}\\\text{からの修正項}}},$$

$$105.4884 \le \underset{\text{満期時価格}}{F_T} \le 108.9907 \quad \text{より}$$

$$Pr\{\underset{(-1\sigma)}{105.4884} \le \underset{\text{満期時価格}}{F_T} \le \underset{(+1\sigma)}{108.9907}\} = \text{約 } 68\% \tag{6.18}$$

尚ルビにある“伊藤の公式からの修正項”に関しては、6.8.2 節で式 (6.51) から式 (6.52) を導く際に説明をしているが、ここでは無視しても良い。[*9]

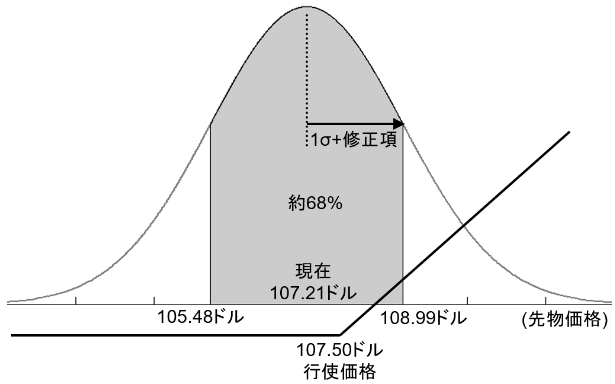


図 6.4: 先物価格の分布とコールオプション

6.2.4 コールオプションの分解と計算イメージ

以上の説明を前提に式 (6.4) がどのような計算を行っているかを見ていこう。ヨーロピアンコールオプションとは満期日に式 (6.19) の金額 (以下、**ペイオフ額**) を受け取る権利のことである。[*10]

コールオプションのペイオフ額 : c_T

$$= \max\big(\text{満期日の先物価格} : F_T - \text{行使価格} : X,\ 0\big) \tag{6.19}$$

[*9] 修正項を無くした場合、1.633%×107.21 ドル=1.75 ドルであり、107.21 ドル± 1.75=105.46 ドルと 108.96 ドル。修正項を入れた式 (6.18) と大差は無い。

[*10] **プットオプション**のペイオフ額は max(行使価格 − 満期日の先物価格, 0)。

アメリカン オプション (**American option**) とは満期日までなら、常時権利行使できるタイプのオプション。

バミューダン オプション (**Bermudan option**) は満期日までの複数の特定日に権利行使できるタイプのオプションで、アメリカン オプションの特殊ケース。本書では「アメリカン」と「バミューダン」を区別する必要がない場合、同意語として使用する。

オプション価格とはこのペイオフ額の期待値の現在価値であり、ここでは現在価値化を省いた“ペイオフ額の期待値”の部分を見ていく。

OTM のペイオフ額はゼロなので、式 (6.19) は ITM の範囲のみを考えることで max 記号を取り外すことができる。**期待値**演算を $E[\cdots]$ で表すと ペイオフ額の期待値は次式のようになる[*11]。

ペイオフ額 (コール) の期待値 (ITM のみ)

$$
\begin{aligned}
&= E\Big[max(F_T - X, 0)\Big] \;=\; E\Big[\; F_T - X \;\Big]_{ITM} \\
&= E\big[\; F_T \;\big]_{ITM} - E\big[\; X \;\big]_{ITM} \\
&= \Big[\overset{ITM\text{ 時の}}{\text{先物価格の期待値}}\Big] \;-\; \Big[\overset{ITM\text{ 時の}}{\text{行使価格の期待値}}\Big] \\
&= \overbrace{\underbrace{F \times N(d_1)}_{\substack{ITM\text{ 時}\\ \text{先物価格 期待値}}} - \underbrace{X \times N(d_2)}_{\substack{ITM\text{ 時}\\ \text{行使価格 期待値}}}}^{\text{ブラック コア}}
\end{aligned}
\tag{6.20}
$$

3 番目の等号はペイオフ額の期待値は 2 つの期待値に分解して計算することを示し、4 番目の等号でそれぞれの期待値に名前を付けた。そして式 (6.20) はブラックコアであり、2 つの期待値の具体的な計算式となっている。

第 1 項: $\boldsymbol{F \times N(d_1)}$: 図 6.1 B17 セルの説明で記した様に $N(d_1)$ はフォワード測度でインザマネーとなる確率であり、それに F を掛けて 先物価格の期待値を算出。

第 2 項: $\boldsymbol{X \times N(d_2)}$: 同図 B18 セルの説明として、$N(d_2)$ がリスク中立確率でインザマネーとなる確率であり、行使価格を掛けて期待値を算出。

ブラックコア : 符号を考慮すれば、“先物価格を受け取り、行使価格を支払う”期待値をブラックコアが計算。

より数式的に展開した説明が以下となるが、積分記号が嫌いな読者は読み飛ばしても構わない。(確率を記述する場合、積分記号は必須アイテムであり、積分の計算はできなくても、意味を理解できるようになろう)

$$
\begin{aligned}
\underset{(\text{満期時})}{c_T} \;&=\; E\Big[max(F_T - X, 0)\Big] \\
&=\; \int_X^{\infty} \underbrace{(F_T - X)}_{\text{ペイオフ額}} \times \underbrace{n(x)}_{\substack{\text{標準正規分布}\\ \text{密度関数}}} \;\; dx \;, \quad x \;=\; \ln F_T
\end{aligned}
\tag{6.21}
$$

[*11] 期待値の演算に関しては各自持っているファイナンスの入門用テキストを参照。 または村上 [9] は数式の展開を親切に記述した良書であり、学習の初期段階や測度の理解で一読をお勧めする。

$$= \int_X^\infty F_T \times n(x)dx - X \underbrace{\int_X^\infty n(x)dx}_{\substack{\text{標準正規分布}\\ \text{分布関数}=N(d2)}} \tag{6.22}$$

式 (6.21)：積分区間を $[X, \infty]$ (ITM のみ) とし、ペイオフ額の期待値を計算する式。

- x は先物価格に対数をとった変数。
- 先物価格は対数正規分布を仮定。対数をとった x は正規分布。

式 (6.22)：第 1 項が先物価格の期待値、第 2 項が行使価格の期待値。

- 第 2 項はこのままで $X \times N(d2)$。
- 第 1 項の計算は複雑であり、村上 [9] 1.6 節または西村 [7] 式 (3.53) 等の記述を参照頂きたい。

6.2.5　ブラックコア算出値の単位

マイナス金利が発生する以前の Tibor/Euribor やスワップレートは対数正規分布が仮定され、Cap/Floor やスワップションは次式 (6.23), (6.24) で計算していた。

(ブラックモデルのスワップション/キャプレット価格式)

ペイヤースワップション (コール)：$\boldsymbol{c}$

$$= \text{アニュイティー} \times \left[\overbrace{\underbrace{F}_{\substack{\text{フォワード}\\ \text{金利}}} \quad N(d_1) - \underbrace{X}_{\substack{\text{行使価格}\\ \text{(クーポン)}}} \quad N(d_2)}^{\text{ブラック コア}} \right] \tag{6.23}$$

$$d_1 = \frac{ln(F/X)}{\sigma\sqrt{T}} + \frac{\sigma\sqrt{T}}{2}、\quad d_2 = d_1 - \sigma\sqrt{T}$$

F：オプション満期 T 年のフォワード (スワップ) 金利

X：行使価格 (対象スワップのクーポン)

キャプレット：$\boldsymbol{c}$(利息期間の日数：t)

$$= \overbrace{\underbrace{D_p}_{\substack{\text{ディスカウント}\\ \text{ファクター}}} \times \frac{t}{360}}^{\substack{\text{1 期間の}\\ \text{アニュイティー}}} \times \left[\overbrace{\underbrace{FN(d_1) - XN(d_2)}_{\text{(インザマネー分の金利)}}}^{\text{ブラック コア}} \right] \tag{6.24}$$

D_p：支払日 p のディスカウントファクター

式 (6.4) も含め、ブラックコア $[FN(d_1) - XN(d_2)]$ は共通に用いられている。上 2 つの式のブラックコアが何を算出しているかを簡単に確認しておく。

スワップション式 (6.23)： 行使価格 X にはスワップレートが与えられることから、コアから計算される数値はスワップレート。 そのレートにアニュイティを掛け、現在価値 (オプション価格) を算出。

キャプレット式 (6.24)： 行使価格 X には Tibor/Euribor 等の金利が与えられることから、コアから計算される数値は Tibor/Euribor の金利。 そのレートに 1 期間のアニュイティを掛け、オプション価格を計算。

マイナス金利の出現以降、これらの金利オプションは正規分布を仮定したノーマルモデル (第 7 章で説明) で計算することが主流となった。 ノーマルモデルにも“ コア ”があり、“ コアの単位 ”を考えることはオプションの計算を理解する上で重要。

6.2.6 エクセル読込みとヒストグラム

では図 6.2 のエクセルを pandas のデータフレームに変換し、図 6.3 の統計値算出とヒストグラム描写のコードを説明する。

```
1 from myABBR import *  # numpy, pandas, matplotlibのみを使用
2
3 # エクセル読込み
4 dfFUT5  = pd.read_excel('ch06.xlsm', 'HistPrc')
5 dfFUT5  = dfFUT5[:-1] ; display(dfFUT5[:2], dfFUT5[-2:])        #最後の1行を削除
6 # 統計値算出
7 dscrRTN = dfFUT5.logRTN.describe(percentiles=[]); display(dscrRTN)
```

	tradeDT	lastPRC	smpRTN	logRTN	StDev5ds	HistVol5ds
0	2023-07-21	107.257812	0.000219	0.000219	0.002855	0.045142
1	2023-07-20	107.234375	-0.005218	-0.005232	0.003008	0.047559

	tradeDT	lastPRC	smpRTN	logRTN	StDev5ds	HistVol5ds
137	2023-01-04	108.843750	0.003024	0.003019	NaN	NaN
138	2023-01-03	108.515625	0.000793	0.000792	NaN	NaN

```
count    139.000000
mean      -0.000078
std        0.004053
min       -0.008147
50%       -0.000362
max        0.012165
Name: logRTN, dtype:
```

図 6.5: 図 6.2 のエクセルの読み込みと統計値算出のコード

図 6.5 はエクセルをデータフレームに変換し、7 行目の describe メソッドで統計値を算出させたコードとなる。

4: `read_excel`： pandas の read_excel 関数にファイル名'ch06.xlsm'、シート名'HistPrc' を与え、データフレーム dfFUT5 を作成。

- 図 6.2 エクセル 1 行目の文字列がデータフレームの列名を構成。

5: スライス `[:-1]`： データフレーム dfFUT5 から、最後の行 (139 行目) を削除させる為のスライス [:-1]。

8: `describe`： logRTN 列の統計値を describe メソッドで算出。

- 引数 **percentiles=[]** は 25%等の**パーセンタイル**値を非表示。
- 出力にある“50%”は**中央値** (**median**) を意味。

図 6.3 のヒストグラムは次のコードで描写させた。

```
1 # ヒストグラム
2 fig, ax = plt.subplots(figsize=(5,2))
3 ax.set_xlabel("収益率") ; ax.set_ylabel("頻度")
4 ax.xaxis.set_major_formatter(mtick.PercentFormatter(1,2))           # x軸 %表示
5 ax.grid(linestyle='--',linewidth=1)                                 # グリッド表示
6 ax.hist(dfFUT5.logRTN,bins=np.linspace(-.015,.015,31),color='c',ec='w',lw=3);
```

図 6.6: ヒストグラム (図 6.3) のコード

2〜5： 2 行目〜5 行目に関しては図 1.12 の説明を参照。

6: ax.hist： hist メソッドでヒストグラムを logRTN 列から作成。

- 引数 **bins=**np.linspace(⋯, 31) はヒストグラムの棒を 31 本で描くことを指示。

6.3 債券先物オプションの QuantLib コード

QuantLib には BlackCalculator というクラスが用意され、前節までで紹介した計算を簡便に行うことができる。初めにこのクラスによる計算例を紹介する。

ただし BlackCalculator クラスはモンテカルロシミュレーション等を行えない為、6.3.2 節で 2 つ目の計算例として、確率過程のオブジェクトを作成する正攻法なコードを説明する。

6.3.1 BlackCalculator クラス

下図はリファレンスマニュアルの **BlackCalculator** コンストラクタ画面で、必要な引数は StrikedTypePayoff, forward, stdDev, discount の 4 つとなる。

BlackCalculator Class Reference

Black 1976 calculator class. More...

Public Member Functions

BlackCalculator (const ext::shared_ptr< **StrikedTypePayoff** > &payoff, **Real** forward, **Real** stdDev, **Real** discount=1.0)

Real **value** () const	
Real **deltaForward** () const	**Real** **vega** (**Time** maturity) const
virtual **Real** **delta** (**Real** spot) const	**Real** **rho** (**Time** maturity) const
Real **elasticityForward** () const	**Real** **dividendRho** (**Time** maturity) const
virtual **Real** **elasticity** (**Real** spot) const	**Real** **itmCashProbability** () const
Real **gammaForward** () const	**Real** **itmAssetProbability** () const
virtual **Real** **gamma** (**Real** spot) const	**Real** **strikeSensitivity** () const
virtual **Real** **theta** (**Real** spot, **Time** maturity) const	**Real** **strikeGamma** () const
virtual **Real** **thetaPerDay** (**Real** spot, **Time** maturity) const	

- ブラックモデル式 (6.3) を離散化した式 (6.25) に対し、手計算で数値例を示せた。(眺めるだけだった方程式 (6.3) に数字を当てはめ、計算でき、方程式の理解が進んだことが重要)
- QuantLib の GaussianPathGenerator クラス (図 6.13 の 9 行目) は適切に離散化を計算できている。

図 6.15 は価格パス mtrxPT を描写したグラフである。グラフ化すると判り易く、満期時に 110 ドルを超えたパスが図 6.14 で計算した mtrxPT[0] である。モンテカルロシミュレーションにおいて、パスのグラフ化は必須ツールであろう。

```
1 # パスのプロット
2 %matplotlib inline
3 fig, ax = plt.subplots(figsize=(4,2))
4 for ii in range(nPATH): ax.plot(timeGRD, mtrxPT[ii])
5 ax.set_title("Black model モンテカルロ パス")
6 ax.set_xlabel("年"); ax.set_ylabel("先物価格")
7 ax.grid(linestyle='--',linewidth=1);
```

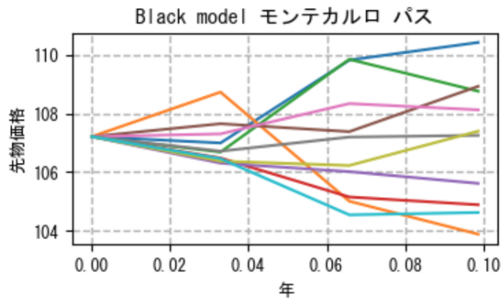


図 6.15: パスの描写

このコードは図 6.5, 6.6 とほぼ同じであり、相違する部分は次の 2 点。

2: `%matplotlib inline`： Jupyter Notebook で Matplotlib のグラフ表示を行う**マジックコマンド** (**Magic Commands**)。

- 他のグラフ表示ではこのコマンドは不要だったが、この価格パスのグラフで必要となった。(理由は不明だが、Jupyter 実行環境の依存が原因かと思われる)

4: for ii in range(nPATH): ax.plot： ax.plot を for ループさせ、10 本のパスを描写。

6.5　BS モデルでのリスク中立法と 2 項ツリー

この 6.5 節では初めに BS モデルで**リスク中立法**をイメージ的に説明し、Q メジャーと T-forward メジャーを導入する。これらのメジャーによって、BS モデルとブラックモデルの相違を明らかにする。

無配当株式 S_t に対する BS モデルの確率微分方程式は μ を投資家 A の期待リターンとすると、

$$\frac{dS_t}{S_t} = \mu dt + \sigma dW_t^P \tag{6.26}$$

と表せ、r をオプション満期時点のリスクフリーレートとすると、リスク中立法によって、

$$\frac{dS_t}{S_t} = rdt + \sigma dW_t^Q \quad \text{(式 6.1に同じ)}$$

に変換される。ここでブラウン運動 W の肩にある $\boldsymbol{P}$ は**観測確率**、$\boldsymbol{Q}$ は**リスク中立確率**を表す。

(P メジャー、期待リターン、上昇/下落確率)

初めにこの観測確率 P について、数値例で考えよう。投資家 A は下図のように現在価格 100 円の株式 S が 1 年後に確率 90%で 110 円になるか、または確率 10%で 90 円になると予想していると仮定する。

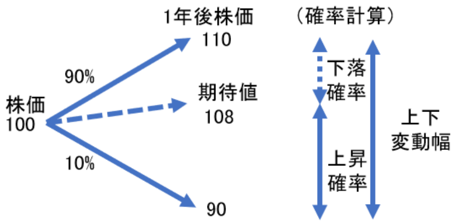


図 6.16: 投資家 A の株価予想と確率の計算イメージ

(確率計算において上昇幅, 下落幅が逆転している点に注意)

この 90%、10%が観測確率 P である。式 (6.26) の dW_t^P とは、ブラウン運動がこのような確率で動いていることを表す為の記号で、それほどの他意はない。

また“確率 90%、確率 10%”のように“確率”という用語は 1 つの数字と結び付く場合に使用されることが多く、まとめて呼ぶ場合“測度”と言うことが多い[*15]。つまり 90%、10%の 2 つの確率を**測度 P**(**P メジャー**, P-measure) として参照する。

この株式の期待値を測度 P で計算すると次式となる。

$$\textbf{期待値}: E^P(S) = 110\,\text{円} \times 90\% + 90\,\text{円} \times 10\% = 108\,\text{円}$$

[*15] 著者の個人的解釈であり、確率/測度の正式な定義では無い。

リスクフリーレートを $r = 5\%$ とし、投資家 A の期待リターン μ は

$$\begin{aligned}\textbf{期待リターン}:\mu \underset{\substack{(\text{式 }6.26\text{の}\\ \text{ドリフト項})}}{} &= \frac{E^P(S)-S}{S} = \frac{108\text{円}-100\text{円}}{100\text{円}} = 8.0\% \\ &= \underbrace{5.0\%}_{\text{リスクフリーレート}:r} + \underbrace{3.0\%}_{\text{リスクプレミアム}}\end{aligned}$$

となる。

一方、期待リターン 8%が与えられた場合、上昇及び下落確率は次にように計算出来る。

$$\textbf{上昇確率}:P_u = \underbrace{\frac{\overbrace{108\text{円}}^{\text{期待値}}-\overbrace{90\text{円}}^{\text{下落価格}}}{110\text{円}-90\text{円}}}_{\text{上下変動幅}} = 90\% \tag{6.27}$$

$$\textbf{下落確率}:P_d = \underbrace{\frac{\overbrace{110\text{円}}^{\text{上昇価格}}-\overbrace{108\text{円}}^{\text{期待値}}}{110\text{円}-90\text{円}}}_{\text{上下変動幅}} = 10\%\ , \tag{6.28}$$

$$(\ P_d = 1-P_u\ ) \tag{6.29}$$

この計算法は図 6.16 から推測して頂きたい。ここでは上下に動く確率と期待リターンが表裏一体になっていることを理解しよう。

(行使価格 105 円コールオプションの価格)

ここで、もし株式 S に対する 105 円のコールオプションが次の計算により価格 4.17 円で売られていたら、投資家 A は購入すべきであろうか？

(コールオプション価格の計算)

1 年リスクフリーレート： 5%， S 株リスクプレミアム： 3%

満期時 ITM の期待値 : $(110\text{円}-105\text{円}) \times \underbrace{90\%}_{\text{上昇確率}} = 4.5\text{円}$

$$\text{コール価格} = \frac{4.5\text{円}}{1+(5\%+3\%)} = 4.17\text{円} \tag{6.30}$$

この計算では満期時点の期待値を株式 S のリスクプレミアムを乗せた割引率 8%で割り引いているので、一見 公正な価格のようである。ただし、オプションのリスクプレミアムは原株式 S よりも大きいことを忘れた評価となっている。

6.5.1　複製ポートフォリオによる評価

複製ポートフォリオの考え方でこの 105 円コールの価格を求めてみる。

株式 S を x 単位買い、リスクフリーレートで B 円の借り入れを行い、105 円コールを複製出来たとしよう。この場合、**複製ポートフォリオ**とコールオプションの満期時価値から、次の方程式を作ることが出来る。

$$
\begin{aligned}
&\text{(現在のポートフォリオ価値)} \\
&\quad 100\,\text{円} \times x\,\text{株} - B\,\text{円} \;=\; \text{コール価値} \qquad (6.31) \\
&\text{(上昇時の価値)} \\
&\quad 110\,\text{円} \times x\,\text{株} - (1+5\%) \times B\,\text{円} \;=\; \underbrace{5\,\text{円}}_{\substack{\text{コール} \\ ITM\,\text{額}}} \qquad (6.32) \\
&\text{(下落時の価値)} \\
&\quad 90\,\text{円} \times x\,\text{株} - (1+5\%) \times B\,\text{円} \;=\; \underbrace{0\,\text{円}}_{OTM\,\text{額}} \qquad (6.33)
\end{aligned}
$$

x と B の 2 つの未知数に対して、上昇と下落の式 (6.32)、(6.33) の 2 つの方程式が得られるので、x と B は簡単に求めることが出き、

$$x \;=\; 0.25\,\text{株}\,, \quad B \;=\; 21.43\,\text{円}$$

を得る。この結果を式 (6.31) 左辺に代入すると、コール価格は 3.57 円となる。

$$\text{コール価格} \;=\; 100\,\text{円} \times 0.25\,\text{株} - 21.43\,\text{円} \;=\; 3.57\,\text{円} \qquad (6.34)$$

この複製ポートフォリオの計算では確率を使用していない点に注意しよう。尚、オプションのリスクプレミアムは式 (6.30) を逆算すると 21%程度が要求されている。

6.5.2 リスク中立法, Q メジャー

正しいコール価格が判ったところで、**リスク中立法**を説明しよう。投資家 A とは別に“株式 S の期待リターンは 1 年リスクフリーレートと変わらない”と考えている投資家 N を仮定する。

この投資家 N が住む世界がリスク中立な世界であり、N は株式のリスクに無頓着でリスクプレミアムを要求しない。従って、投資家 N の株式 S の期待値はリスクフリーレートから 105 円となり、110 円に上昇する確率、90 円に下落する確率は次のように計算出来る。 (式 6.27, 6.28 比較。)

$$\textbf{上昇確率} : \boldsymbol{Q_u} \;=\; \frac{\overbrace{105\,\text{円}}^{\text{期待値}} - \overbrace{90\,\text{円}}^{\text{下落価格}}}{\underbrace{110\,\text{円} - 90\,\text{円}}_{\text{上下変動幅}}} \;=\; 75\% \qquad (6.35)$$

$$\textbf{下落確率} : \boldsymbol{Q_d} \;=\; \frac{\overbrace{110\,\text{円}}^{\text{上昇価格}} - \overbrace{105\,\text{円}}^{\text{期待値}}}{\underbrace{110\,\text{円} - 90\,\text{円}}_{\text{上下変動幅}}} \;=\; 25\% \qquad (6.36)$$

先ほど 投資家 A が間違えて計算した式 (6.30) をリスク中立な投資家 N に対して適応すると、

リスク中立法のコールオプション価格の計算

$$\text{満期時 } ITM \text{ の期待値} : (110\text{ 円} - 105\text{ 円}) \times \underbrace{75\%}_{\text{上昇確率}} = 3.75\text{ 円}$$

$$\text{コール価格} = \frac{3.75\text{ 円}}{1+5\%} = 3.57\text{ 円} \tag{6.37}$$

と正しいコール価格 (式 6.34 での計算) が算出される。

従って オプションの価格を計算するには、期待リターン μ など考えず、始めから

期待リターン μ ＝ リスクフリーレート r

として計算すれば、十分だと考えるのがリスク中立法である。

ただし、リスク中立法を使うと、上昇・下落の確率が観測確率と異なることになるので、式 (6.35)、(6.36) の確率を**リスク中立確率** (**Q メジャー**, Q-measure) と呼び、P メジャーとは区別する。

以上の説明を基に再度、BS モデルの確率微分方程式 (6.1) を見ると、 式 (6.26) の μ を r に、dW_t^P を dW_t^Q に修正したことが理解できるであろう。また、ボラティリティに変更が無い点も注意しておこう。

次に両式の左辺はいずれも $\frac{dS}{S}$ で同じ株価の収益率の変動を表しているので、右辺同士を等号で置き、整理すると

$$\underbrace{rdt + \sigma dW_t^Q}_{\text{式 (6.1)}} = \underbrace{\mu dt + \sigma dW_t^P}_{\text{式 (6.26)}} \text{ より},$$

$$dW_t^Q = \underbrace{\frac{\mu - r}{\sigma}}_{\substack{\text{リスクの}\\\text{市場価格}}} dt + dW_t^P \tag{6.38}$$

を得る。つまり、左辺のリスク中立確率とは観測確率から、時間当たりの**リスクの市場価格**をずらした確率である。式 (6.38) は**ギルサノフの定理**[*16] と呼ばれる。

ここでは式 (6.38) を“ W^P と W^Q では平均がズレている ”だけで、そのズレを直してやると、W^P と W^Q は重なり、同じ正規分布となることを理解すれば良い。

[*16]式 (6.38) でルビを振った“ リスクの市場価格 ”はポートフォリオ理論や経済学にも登場する有名な概念であるが、本書では扱わない。リスクの市場価格やギルサノフの定理の詳細は西村 [7], 村上 [9] 等を参照。

6.4 節ではリスク中立確率でモンテカルロシミュレーションを行ったが、式 (6.38) は標準正規分布から乱数を取り出す部分では W^P と W^Q を区別する必要がないことを示している。(または測度の違いはドリフト項に現れる)

6.5.3 BS モデルとブラックモデルの違い、BSM モデル

この 6.5.3 節ではギルサノフの定理 (式 6.38) を使い、BS モデル (式 6.1) とブラックモデル (式 6.3) の違いを説明する。まず リスクフリーレート r で運用される**銀行預金** (**Bank account**) の確率過程を

$$\frac{dB_t}{B_t} = rdt \tag{6.39}$$

で定義する。(この方程式を B_t について解けば、$B_t = e^{rt}$ で通常の複利式)

次に BS モデルで P メジャーの式 (6.26) において、スポット価格を銀行預金で割った確率過程 $d(S_t/B_t)$ は伊藤の商公式 (式 6.69) より 次式となる。

$$\frac{d(S_t/B_t)}{(S_t/B_t)} = (\mu - r)dt + \sigma dW_t^P$$

式 (6.38) を使い、dW_t^P を dW_t^Q に変換すると、

$$\frac{d(S_t/B_t)}{(S_t/B_t)} = \sigma dW_t^Q \tag{6.40}$$

のようにドリフト項が無い式を得る。(ドリフトの無い確率過程のことを**マルチンゲール**、Martingale と言う)

数式の表示法は異なるが、この式 (6.40) は BS モデル (式 6.1) と同じ式である。[*17]村上 [9] はこの辺りの計算を優しく説明した良書で、" **割ったらマルチンゲール** " という親しみ易いフレーズを多用しているが、この場合 銀行預金で割ったスポット価格過程はマルチンゲールになっている。この " 割る資産 " を**ニューメレール** (**基準材**, Numeraire) と呼ぶ。

一方、ブラックモデルの式 (6.3) は $F_t = S_t/P_{(t,T)}$ なので (式 6.2 参照) 、T 年満期の割引債がニューメレールとなったスポット価格の確率過程を次式のように表している。

$$\frac{dF_t}{F_t} = \frac{d(S_t/P_{(t,T)})}{(S_t/P_{(t,T)})} = \sigma dW_t^T \tag{6.41}$$

リスク中立確率に対比させ、T 年満期の割引債をニューメレールとした確率を**フォワード中立確率** (**T-forward メジャー**, T-forward measure) と呼ぶ。

[*17] 式 (6.40) から式 (6.1) を得るには $(S_t/B_t) \times B_t$ という確率過程を考えればよく、伊藤の積公式 (式 6.68) から簡単に式 (6.1) を導出可能。

従って、式 (6.40) と式 (6.41) の相違が BS モデルとブラックモデルの違いとなり、次のようにまとめておく。

BS モデル： ニューメレールは銀行預金で、リスク中立測度で表したスポット価格の確率過程のモデル。

- モデル式は式 (6.40) ではなく、式 (6.1) で表記。

ブラックモデル： ニューメレールは T 年満期の割引債で、T-forward メジャーで表したスポット価格の確率過程のモデル。

- フォワード価格は式 (6.2) の $F_t = S_t/P_{(t,T)}$ と表せる為、フォワード F_t のままでモデル式を記述。

以上 抽象的な概念を使って簡潔に説明したが、ここでは“ 2 つのモデルの出発点は同じ式 (6.26) であり、測度の違い (Q メジャーと T-forward メジャー) から 2 つのモデル式が生まれている ”点を理解すればよい。(T-forward メジャーは金利が変動する第 9 章で準主役となる)

(汎用ブラック ショールズ マートン価格式)

次に各モデルのオプション価格式を比較しておく。[*18]

BS モデルで無配当の株式 S のコールオプション価格 c (プット価格 p) は次式で算出される。

$$
\begin{aligned}
c &= SN(d_1) - e^{-rT}XN(d_2) \qquad (6.42)\\
p &= -\big[SN(-d_1) - e^{-rT}XN(-d_2)\big] \qquad (6.43)\\
&\quad d_1 = \frac{ln(S/X) + rT}{\sigma\sqrt{T}} + \frac{1}{2}\sigma\sqrt{T}、\qquad d_2 = d_1 - \sigma\sqrt{T}\\
&\quad S : \text{株式価格}\quad (\text{その他は式 6.4の定義を参照})
\end{aligned}
$$

1973 年 Robert C. Merton は配当のある株式にも対応出来るように BS モデルを拡張したが、そのモデルは **BSM モデル**と呼ばれる。

オプション満期までの**キャリーコスト *b*** を“ リスクフリーレートと配当利回りの差 ”として定義した場合、ブラックモデル, BSM モデル, 通貨オプショ

[*18] ここの数式は Haug [31] 7 ページを参考にした。

ンモデル等のオプション価格式は統一的に式 (6.44) で表される。[*19]

$$
\begin{aligned}
c \;&=\; e^{(b-r)T}SN(d_1) - e^{-rT}XN(d_2) \qquad (6.44)\\
& d_1 = \frac{ln(S/X)+bT}{\sigma\sqrt{T}} + \frac{1}{2}\sigma\sqrt{T}、\qquad d_2 = d_1 - \sigma\sqrt{T}\\
& q: \text{配当利回り} \qquad r: \text{リスクフリーレート}\\
& b: \text{キャリーコスト} \;=\; r - q
\end{aligned}
$$

b の値に応じ、式 (6.44) は以下のように各モデルのオプション価格式となる。

$\boldsymbol{b = 0}$ **:** 配当利回り q とリスクフリーレート r が等しい場合、式 (6.44) は $e^{-rT}[SN(d_1) - XN(d_2)]$ となり、ブラック価格式 (6.4) となる。

$\boldsymbol{b = r}$**:** 無配当 ($q = 0$) の場合が $b = r$ であり、BS モデルの価格式 (6.42) が計算される。

$\boldsymbol{b = r - q}$**:** マートンが拡張した配当利回り q の価格式。

- 図 6.18 で使用する BlackScholesMertonProcess クラスのオプション価格式。
- リファレンスマニュアルの BlackScholesMertonProcess Class には BSM モデルの確率微分方程式が掲載されている。(下図)

Detailed Description

Merton (1973) extension to the Black-Scholes stochastic process.

This class describes the stochastic process ln(S) for a stock or stock index paying a continuous dividend yield given by

$$d\ln S(t,S) = (r(t) - q(t) - \frac{\sigma(t,S)^2}{2})dt + \sigma dW_t.$$

$\boldsymbol{b = r - r_f}$ **:** q に外国のリスクフリーレート r_f を与えると、1983 年 M. Garman と S. Kohlhagen によって発表された通貨オプション用モデル (**ガーマン コヘーゲン モデル**, Garman Kohlhagen model) の価格式となる。

式 (6.44) は**汎用ブラック ショールズ マートン オプション価格式** (Generalized Black-Scholes-Merton option pricing formula と呼ばれる。

6.6　2 項ツリーとアメリカンオプション

6.5 節では BS モデルとブラックモデルの違いが原資産の違い (株、債券等) では無く、単にメジャーの違いである点を説明した。この節では株式 (債券ではなく) を例にして、アメリカンオプションの計算法と QuantLib のコードを紹介する。

[*19] プット価格式は式 (6.43) のようにコール式の全体に -1 を掛けることと標準正規分布の $N()$ の中を $-d_1, -d_2$ と修正することで計算する。

アメリカンオプションの計算は 2 項ツリーでの説明が判り易い為、ヨーロピアンオプションの 2 項ツリーから始めよう。

6.6.1　2 項ツリー (CRR 法) の計算法

BS モデル式 (6.1) で記述される株価過程と整合性を持った **2 項ツリー** (**binomial tree**) は次式で算出される。

$$\text{上昇率}: \ u \ = \ \sigma\sqrt{\Delta t}\,, \quad \text{下落率}: \ d \ = \ -\sigma\sqrt{\Delta t} \ = \ 1/u \qquad (6.45)$$

$$\text{上昇確率}: \ p \ = \ \frac{1}{2} + \frac{r-\frac{1}{2}\sigma^2}{2\sigma}\sqrt{\Delta t}\,, \quad \text{下落確率}: \ 1-p \qquad (6.46)$$

この計算法は Cox *et al.* [24] で発表され、**CRR 法** (**Cox-Ross-Rubinstein Convention**) と呼ばれている[*20]。2 項ツリーで多用される計算法であり、上式の導出はオリジナル論文 [24] を参照。

ここでは以下の条件のトイモデル (toy model) で計算法を確認する。

B17 セル:	現在株価：	100 円	D2 セル:	オプション満期：	1 年
D5 セル:	行使価格：	100 円	D6 セル:	ボラティリティ：	20%
D7 セル:	金利：	5%			

図 6.17 はこの条件の下、CRR 法によってオプション満期までの株価の分布とヨーロピアン、アメリカン両方のプットオプション価格を算出した。

B18 セル: ヨーロピアン：6.179 円
J18 セル: アメリカン　：6.511 円

まず A 列〜I 列までのヨーロピアンオプションの各セルの計算を見て行く。

(上段 7 行目までの各パラメータ計算: D 列と H 列)

D3: ステップ数=3：オプション満期までのツリーの “**枝分れ**回数”。

D4: dt=0.3333：オプション満期 1 年をステップ数で割り、dt を設定。

H2: フォワード倍率 1.0168 $= e^{5\%\times 0.3333\,\text{年}}$：1 円の 1 ステップ後の価値。

H3: 上昇率 1.1224 $= e^{20\%\times\sqrt{0.3333}}$：式 (6.45)。

H4: 下落率 0.8909 $= 1/$上昇率：式 (6.45)。

H6, 7: 上昇確率 0.5433, 下落確率 0.4567：

[*20] 上昇確率 (式 6.46) に関しては式 (6.36) が使われる場合もある。

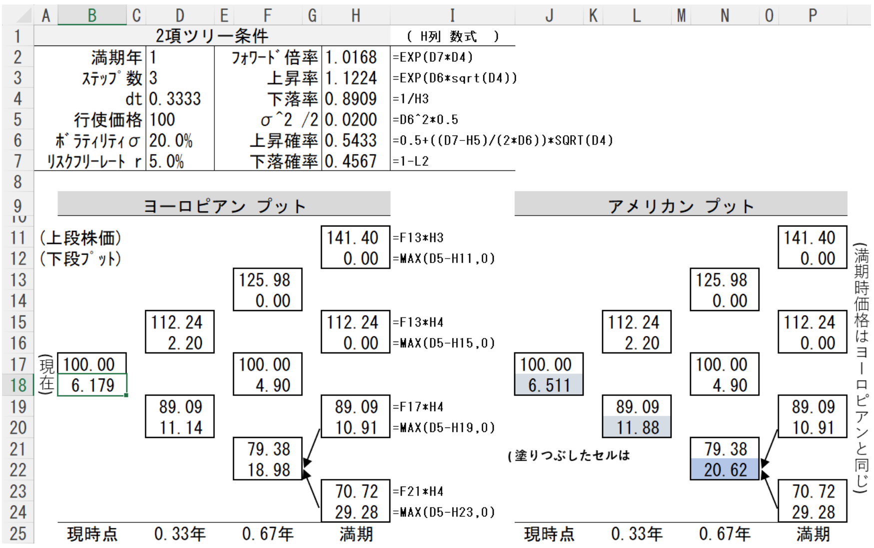


図 6.17: 2 項ツリーでのヨーロピアン プットとアメリカン プットの評価

- 式 (6.46) より 次式で算出。

$$
\begin{aligned}
H6\,\text{セル} &= \frac{1}{2} + \frac{5\% - \frac{1}{2} \times 20\%^2}{2 \times 20\%}\sqrt{0.3333} = 0.5433 \\
H7\,\text{セル} &= 1 - H6\,\text{セル} = 0.4567
\end{aligned}
$$

- 式 (6.1) のボラティリティσ が定数の為、すべての枝分れの上昇/下落確率は同じ。

(株価ツリー作成)

ボックス内の数値：各時点における株価 (上段) とオプション価格 (下段)。

- ボックスを**ノード** (**node**) と呼ぶ。
- 現在のセル (B17, 18) を原点と呼ぶ。

D15、19: 原点から 1 回目の枝分れ後の株価：

H3、H4 セルの“ 上昇率、下落率 ”の値を使用し、1 回目の枝分れ後の株価を次式で算出。2 回目以降も同様な計算を繰り返す。

$$
\begin{aligned}
D15\,\text{セル} &= \underbrace{\text{ノード株価}}_{(100\,\text{円})} \times \underset{(1.1224)}{\text{上昇率}} = 112.24\,\text{円} \\
D19\,\text{セル} &= \underbrace{\text{ノード株価}}_{(100\,\text{円})} \times \underset{(0.8909)}{\text{下落率}} = 89.09\,\text{円}
\end{aligned}
$$

F、H 列: 2 回目、3 回目の枝分れ後の株価：各ノード株価に上昇率、下落率を掛けることで株価のツリーを作成。

(バックワードによるヨーロピアン プット価格の算出)

ツリーでのオプション評価法は満期時点を始点とし、時間を 1 ステップづつ遡りながら評価を行っていく**バックワード Backward** の手法を用いる。

H24、H20 等: 満期時オプション価値 :

- H23 セルは 3 回下落した 70.72 円の株価。この満期時プットオプションの価値は

$$\underset{(H24)}{29.28\text{ 円}} = \underbrace{100\text{ 円}}_{\text{行使価格}} - \underset{(H23)}{70.72\text{ 円}}$$

- 1 回下落した株価 89.09 円 (H19 セル) のプットオプションの価値 (H20 セル) は

$$\underset{(H20)}{10.91\text{ 円}} = \underbrace{100\text{ 円}}_{\text{行使価格}} - \underset{(H19)}{89.09\text{ 円}}$$

- 株価が上昇した H11、H15 は OTM となり、オプション価値はゼロ。

F22 : 0.67 年時点ヨーロピアンオプション価値 18.98 円 :

- 1 ステップ バックワードさせた F22 セルは 2 回下落した株価 79.38 円でのプットオプションの価値を算出。
- ヨーロピアン プットではこの時点の行使が出来ないので、このノードにおけるオプション価値は株価 79.38 円と無関係。
- F21 セルのノードから進むことが出来るノードは H19、H23 であり、それぞれのノードに到達できる確率は H6, H7 セルで計算した上昇/下落確率。
- このノード (F22 セル) でのオプション価値は次式の期待値で算出。

$$\overbrace{\underset{(F22)}{18.98\text{ 円}}}^{\substack{\text{オプション}\\\text{継続価値}}} = \left\{ \underbrace{\underset{(H20)}{10.91\text{ 円}} \times \overbrace{\underset{(H6)}{0.5433}}^{\text{上昇確率}} + \underset{(H24)}{29.28\text{ 円}} \times \overbrace{\underset{(H7)}{0.4567}}^{\text{下落確率}}}_{\text{0.67 時点から見たオプション満期時の期待値の計算}} \right\} \div \overbrace{\underset{(H2)}{1.0168}}^{\substack{\text{1 期間}\\\text{ディスカウント}}} \tag{6.47}$$

- もし 行使が出来るのなら、このノードのオプション価値は 20.62 円 (=100-79.38)。この価値を**行使価値** (または**本源的価値**) と呼ぶ。
- 一方、上式で計算した価値をオプションの**継続価値**と呼ぶ。
- “ 継続価値 ” とはオプションを行使せずに保有を続けた場合の期待値を意味。(“ 行使価値 ” の対義語)

B18: オプション価格 6.179 : 各ノードで式 (6.47) を計算し、バックワードさせることで B18 セルのオプション価格を算出。

(アメリカンオプション評価での相違点)

次に J 列〜P 列のアメリカンオプションの計算をヨーロピアンとの比較で見て行く。

P 列のオプション価値：この値はヨーロピアンの H 列と同じ。

N22: 0.67 年時点のアメリカンオプション価値 20.62 円：

F22 セルで説明したように、このノードでは継続価値よりも行使価値が大きい為、オプション価値は行使価値 20.62 円 (=100-79.38)。

L20: 0.33 年時点のアメリカンオプション価値 11.88 円：

0.67 年の N22 セルでオプション価値 20.62 円となった為、バックワードした L20 セルもヨーロピアンより高い価値。

- このノードでの行使価値は 10.91 円 (=100-89.09)。
- 継続価値は式 (6.47) と同様な式で算出され 11.88 円。
- 従って、このノードのオプション価値は継続価値の 11.88 円。

J18: 現時点のアメリカンオプション価格 6.5118 円：

N22 セルで行使価値となった為、ヨーロピアンよりも 0.34 円程 高い価格。

6.6.2 QuantLib の 2 項ツリーコード

エクセルで計算した図 6.17 のヨーロピアンオプションの価格を QuantLib のコード (図 6.18) で再現させた。両者のヨーロピアンプット価格 6.179 円が一致している点を確認しよう。

```
1  from myABBR import *    ; import myUtil as mu
2  tradeDT = jDT(2024,3,1) ; setEvDT(tradeDT)
3  matDT   = calNL.advance(tradeDT, 1, YY)
4
5  spotPRC, strkPRC, volRT, rfRT, divRT, pC, nSTEP, nPATH, nSeed, mtrxPT   =¥
6  100,       100,     0.20,   0.05,   0.0,   -1,    3,     8,       1,    []
7                                    # pC:put=-1                 mtrxPT:各パス保存
8  # BSM process
9  spotHDL       = mu.sqHDL  (spotPRC)
10 rfOBJ,rfHDL   = mu.ffTSH  (tradeDT,  rfRT,        dcA365)
11 _,    divHDL  = mu.ffTSH  (tradeDT, divRT,        dcA365)
12 volHDL        = mu.bVolTSH(tradeDT, volRT, calNL, dcA365)
13 bsmPROC  = ql.BlackScholesMertonProcess(spotHDL, divHDL, rfHDL, volHDL)
14
15 # option OBJ
16 payOFF = ql.PlainVanillaPayoff(pC, strkPRC)
17 optOBJ = ql.VanillaOption(payOFF, ql.EuropeanExercise(matDT))
18
19 # CRR tree
20 crrENG = ql.BinomialVanillaEngine(bsmPROC, "crr", nSTEP)
21 optOBJ.setPricingEngine(crrENG)
22 print (f'(European) CRR Tree: {optOBJ.NPV():.5f}(ｽﾃｯﾌﾟ数={nSTEP:})', end='   ')
23
24 # analytic
25 anaENG = ql.AnalyticEuropeanEngine(bsmPROC)
26 optOBJ.setPricingEngine(anaENG)
27 print (f'Analytic: {optOBJ.NPV():.5f}')
```

```
(European) CRR Tree: 6.17935(ｽﾃｯﾌﾟ数=3)   Analytic: 5.57353
```

図 6.18: 2 項ツリー (CRR 法) ヨーロピアンプットのコード

次に 13 行目の **BlackScholesMertonProcess** は BSM モデルの確率過程を作成するコンストラクタである。図 6.8 の 4 行目の BlackProcess 過程は Black モデル用であるが、2 つのコンストラクタは配当の有無 (divHDL) を除き、同じ引数となる点を意識しよう。(BSM モデルは “汎用ブラック ショールズ マートン価格式” 210 ページを参照。以下では BSM モデルを “BS モデル” として参照する)

では図 6.8 と比較しながら、図 6.18 のコードを説明する。

3: matDT=calNL.advance(tradeDT, 1, YY)： calNL の advance メソッドによって、1 年後の応当日をオプション満期日に設定。

- calNL=ql.**NullCalendar**() は myABBR モジュールで定義。NullCalendar は休日を持たないカレンダーオブジェクト。(トイモデルの為、休日は無視)
- **YY**=ql.**Years** も myABBR モジュールで定義した短縮形。

5, 6: 各種パラメータの設定：

- divRT=0.0 は配当レート用変数を設定。
- nPATH=8, nSeed=1, mtrxPT=[ ] はモンテカルロ法で後程使用。(図 6.13 で同様な変数を設定)

9〜12: 各種ハンドルの設定： 13 行目の BlackScholesMertonProcess コンストラクタに与える 4 つの引数を準備。

- 11 行目 divHDL は配当率のハンドルで、その作成はリスクフリーレートと同様、mu.ffTSH を使用。
- 他 3 つのハンドルに関しては図 6.7 の 1 番セル、図 6.8 を参照。

13: `BlackScholesMertonProcess`： BlackScholesMertonProcess コンストラクタの引数は initialValue, dividendTS, riskFreeTS, volTS の 4 つ。

```
ql.BlackScholesMertonProcess(initialValue, dividendTS, riskFreeTS, volTS)
```

- Black コンストラクタの 3 つの引数に dividendTS が追加されている。

16, 17: `PlainVanillaPayoff`、`VanillaOption`： この 2 つのコンストラクタは図 6.7 の 2 番セル及び図 6.8 の 7 行目で説明済み。(図 6.9 も参照)

20: `BinomialVanillaEngine`： 2 項ツリー用エンジンのコンストラクタで引数は最低 3 つ process, type, steps が必要。

process： 13 行目で作成した bsmPROC を指定。

type： ツリー作成法として crr を指定。

steps： ステップ数として 6 行目の nSTEP=3 を指定。

- 例えば、ステップ数を 250(1 年間の毎営業日のイメージ) にした場合、出力は“5.56567(ステップ数=250) ”となり、解析解 (Analytic) に近い評価が可能。

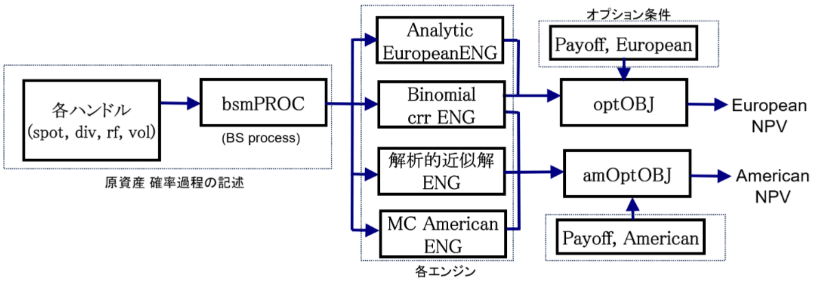
25, 26: anaENG： BS モデルのオプション価格式 (analytic) で評価を行う為に AnalyticEuropeanEngine によって anaENG を作成し、 optOBJ オブジェクトに設定。

- AnalyticEuropeanEngine に関しては図 6.8 の 5 行目参照。
- BS モデルのオプション価格式は??節を参照。

6.6.3　アメリカンオプションの評価

アメリカンオプションの評価は主にツリー、有限差分法、モンテカルロ法、解析的近似解等で行われる。

図 6.18 の設定を前提としたアメリカンオプションを計算するコードが図 6.19 であり、ツリー、解析的近似解、モンテカルロ法の 3 種類の計算例を示した。[*21]




```
1  # American用オブジェクト
2  amOptOBJ = ql.VanillaOption(payOFF,ql.AmericanExercise(tradeDT,matDT))
3
4  # tree
5  amOptOBJ.setPricingEngine(crrENG)
6  print (f'ツリー: {amOptOBJ.NPV():.5f} (ｽﾃｯﾌﾟ数={nSTEP:})')
7
8  # 2つの解析的近似解
9  bawENG = ql.BaroneAdesiWhaleyApproximationEngine(bsmPROC)
10 amOptOBJ.setPricingEngine(bawENG)
11 print (f'近似解: {amOptOBJ.NPV():.5f} (Barone Adesi Whaley)', end='  ')
12
13 bjsENG = ql.BjerksundStenslandApproximationEngine(bsmPROC)
14 amOptOBJ.setPricingEngine(bjsENG)
15 print (f'{amOptOBJ.NPV():.5f} (Bjerksund Stensland) ')
16
17 # MC
18 MCamENG = ql.MCAmericanEngine(bsmPROC, 'lowdiscrepancy', seed=nSeed,
19                 timeSteps=nSTEP, requiredSamples=nPATH, nCalibrationSamples=nPATH)
20 amOptOBJ.setPricingEngine(MCamENG)
21 print (f'MC:     {amOptOBJ.NPV():.5f} (ﾊﾟｽ数={nPATH:})')
```

```
ツリー: 6.51105 (ｽﾃｯﾌﾟ数=3)
近似解: 6.09762 (Barone Adesi Whaley)  5.98297 (Bjerksund Stensland)
MC:     5.50430 (ﾊﾟｽ数=8)
```

図 6.19: アメリカンオプションの各種エンジンによる計算例

ツリーのコードの説明から始め、解析的近似解を補足する。モンテカルロ法は 6.7 節で説明する。

(ツリー: 1〜6 行目)

まず、6 行目 print 文で出力された " tree:6.51105(ステップ数=3) " がエクセルのツリー (図 6.17 の J18 セル) で算出したアメリカンプットと同じ値となっている点を確認しよう。(つまり、QuantLib のツリーの計算は図 6.17 の計算に同じ)

2: amOptOBJ ： アメリカンオプションのオブジェクトを作成。

[*21] 本書では紙面の都合上、**有限差分法** (Finite difference method) を説明しないが、QuantLib では `FdBlackScholesVanillaEngine` コンストラクタ等が提供されている。

VanillaOption : VanillaOption コンストラクタの 2 番目の引数に AmericanExercise を指定することでアメリカンオプションとなる。(193 ページの " 水戸黄門と助さん格さんのイメージ ")

AmericanExercise : 図 6.9 参照。(格さんの衣装がヨーロピアンとは異なっているイメージ)

5: crrENG : このエンジンは図 6.18 の 20 行目で作成済み。

- nSTEP=3 のアメリカンオプションとなっていて、1 年の内 行使できるのは 3 回のみ。
- 毎日行使できるように図 6.18 の 5 行目で nSTEP=250 とした場合、" ツリー: 6.08728 " の評価となる。(各自確認)

(解析的近似解: 8〜15 行目)

アメリカンオプションの解析解は存在しないが、**解析的近似解** (Approximation) は提供されている。図 6.19 の 9 行目と 13 行目でそれぞれ **Barone-Adesi-Whaley** 法と **Bjerksund-Stensland** 法のエンジンを設定し、計算結果をプリントした。

- 先ほどのツリーで nSTEP=250 にした場合の 6.08728 円と 2 つの近似解は似た値を算出していることを確認しよう。
- この 2 つの近似解の計算式は有名な Haug[31] に数値例付きの解説が記載されているので、ここでは省略する。

(図 6.19 のオブジェクトに関して)

ここまでを纏める為、図 6.13 と図 6.19 の 2 つのオブジェクト図を比較しよう。後者はアメリカンオプションを計算させる為にオブジェクトの数は増えているが、基本的な構成に大差はない。(4 番目の MCAmericanENG エンジンは 6.7 節で説明)

大まかに言えば、確率過程の記述、各種エンジン、そしてオプションオブジェクトから NPV を算出している。

6.7 最小二乗モンテカルロ法 (LSM)*

2001 年以前一部の研究を除き、モンテカルロ法でアメリカンオプションを計算させることは出来ないと考えられていた。その理由は図 6.17 で説明したようにアメリカンオプションでは継続価値と行使価値を比較する必要があるが、モンテカルロ法での継続価値の算出法が不明だった為である。ツリーで

は式 (6.47) で示したようにノードから分岐する確率を使い、簡単に継続価値を計算できていた。

しかし 2001 年 Longstaff と Schwarts が発表した論文 [40] によって、アメリカンオプションをモンテカルロ法で計算する方法が確立した。その計算方法では最小二乗法を使用する為、**最小二乗モンテカルロ法** (Least Square Monte Carlo Methods, 以下 **LSM** と参照) と呼ばれる。[*22]

最小二乗法 (**回帰分析**としても参照) とは**回帰式** (6.48) において X を**説明変数**, Y を**従属変数** (被説明変数) として Y と最小二乗推定値 $\hat{Y}$ の差の 2 乗和が最小となる a, b, c を求める方法である。

$$\underbrace{Y}_{\text{従属変数}} \approx \underbrace{\hat{Y}}_{\substack{\text{最小 2 乗}\\ \text{推定値}}} = a \underbrace{X^2}_{\substack{\text{説明変数}\\ \text{の 2 乗}}} + b \underbrace{X}_{\text{説明変数}} + c \tag{6.48}$$

論文 [40] ではモンテカルロシミュレーションで使用した株価を説明変数に、**同一パスにある次のステップ以降の行使価値**を従属変数として回帰分析し、算出された最小二乗推定値 (以下 **回帰線**と参照) を継続価値としている。

この有名な論文 [40] は優しい数値例での説明となっているので、一読すべきである (最低 数値例の部分だけでも良い) 。また多くのテキストで同様な数値例が記載されている為[*23] 、本書では最小二乗モンテカルロ法をコーディングと関連付けて説明する。

6.4 節で MCEuropeanEngine(図 6.11) の計算結果をメルセンヌ・ツイスタ法の疑似乱数で再現したように、MCAmericanEngine というエンジンの計算結果 (図 6.19 の 18 行目のエンジンによる 5.5043) をソボル列と LSM の手計算で再現する。

LSM は最も代表的なアメリカンオプションの計算法であり、特に複数資産やパス依存の商品を評価する際の必須アイテムとなっている。この節は多少長い説明となる。

6.7.1　LSM の QuantLib コード

まだ説明をしていない図 6.19 の 18 行目に戻ろう。MCAmericanEngine エンジンの設定に関しては図 6.11 の MCEuropeanEngine コンストラクタとほぼ同じ仕様となり、難しくはない。

[*22] **アメリカン モンテカルロ** (American Monte Carlo) や Regression based Monte Carlo 等とも言われる。

[*23] 例えば Hull[34] では Longstaff,Schwarts[40] と同じ数値例で説明。

18〜20: MCamENG： 最小二乗モンテカルロ用エンジンのオブジェクト MCamENG を 18 行目で作成し、アメリカンオプション用オブジェクト amOptOBJ にセット。

18: MCAmericanEngine： 下図が最小二乗モンテカルロ用エンジンのコンストラクタの仕様。

```
ql.MCAmericanEngine(GeneralizedBlackScholesProcess, traits, timeSteps=None, timeStepsPerYear=None,
antitheticVariate=False, controlVariate=False, requiredSamples=None, requiredTolerance=None,
maxSamples=None, seed=0, polynomOrder=2, polynomType=0, nCalibrationSamples=2048,
antitheticVariateCalibration=None, seedCalibration=None)
```

- 必要な引数は GeneralizedBlackScholesProcess、traits、timeSteps、requiredSamples の 4 つ。
- この 4 引数は MCEuropeanEngine と同じであり、ここでは注意すべき引数のみを説明。(その他は図 6.11 の説明を参照)

第 1 引数: `GeneralizedBlackScholesProcess`： このコンストラクタで指定できる確率過程は??節で説明したブラックモデルや BS モデル等のみとなる。

- 第 7 章ノーマルモデルや第 9 章 Hull-White モデルは対応していない点に注意。

第 2 引数: `traits`： この例図 6.19 では **`lowdiscrepancy`** を指定し、**ソボル列**を使った準モンテカルロ法とした。

- low discrepancy(ソボル列)、準モンテカルロ法に関しては図 6.11 の traits の説明を参照。

オプショナルな引数: `nCalibrationSamples`=nPATH： `nCalibrationSamples` の初期値はコンストラクタの仕様で 2048 であるが、nPATH に変更し、手計算の結果と一致させた。

20: amOptOBJ.NPV： 3 ステップ, 8 パスで出力された NPV は 5.5043 円。

- 250 ステップ, 2000 パスにした場合、“ MC: 5.97636 ” と解析的近似解に近い値となる。(各自確認)

6.7.2　準モンテカルロ法での価格パス作成

ここでは準モンテカルロ法で株価パスを作成する。図 6.20 がそのコードとなるが、新しい部分は 15〜18 行目であり、その他は説明済みのコードからのコピーである。初めにコピー元の図と行番号を示しておく。

```
 1 from myABBR import *    ; import myUtil as mu
 2 tradeDT = jDT(2024,3,1) ; setEvDT(tradeDT)
 3 matYR   = 1 ; matDT  = calNL.advance(tradeDT, matYR, YY)
 4 
 5 spotPRC, strkPRC, volRT, rfRT, divRT, pC, nSTEP, nPATH, nSeed, mtrxPT   =¥
 6 100,       100,     0.20,   0.05,   0.0,   -1,    3,      8,      1,      []
 7                                # pC:put=-1                    mtrxPT:各パス保存
 8 # BSM process
 9 spotHDL      = mu.sqHDL  (spotPRC)
10 rfOBJ,rfHDL = mu.ffTSH  (tradeDT,  rfRT,        dcA365)
11 _,    divHDL = mu.ffTSH  (tradeDT, divRT,        dcA365)
12 volHDL       = mu.bVolTSH(tradeDT, volRT, calNL, dcA365)
13 bsmPROC      = ql.BlackScholesMertonProcess(spotHDL, divHDL, rfHDL, volHDL)
14 
15 # ソボル列からパスを生成
16 uniSeqRNG = ql.UniformLowDiscrepancySequenceGenerator(nSTEP, nSeed)
17 gsSeqRNG  = ql.GaussianLowDiscrepancySequenceGenerator(uniSeqRNG)
18 pathGEN   = ql.GaussianSobolPathGenerator(bsmPROC, matYR, nSTEP, gsSeqRNG, False)
19 
20 # 各パスをmtrxPTへ保存 (nextメソッドで乱数を発生、valueで値を取得)
21 for xSTK in range(nPATH):
22     onePT = pathGEN.next().value()
23     mtrxPT.append(list(onePT))                       # mtrxPTはnPATH行, nSTEP+1列
24 # onePTから時間軸を作成 (timeメソッドで時間軸の値を取得)
25 timeGRD = nA([onePT.time(i) for i in range(len(onePT))])
26 mtrxPT = nA(mtrxPT) ; print('<<mtrxPT>>¥n',mtrxPT)            # mtrxPTはnAに変更
27 
28 # パスのプロット(nSTEP<5で描写)
29 if nSTEP<5:
30     fig, ax = plt.subplots(figsize=(4,2))
31     for i in range(nPATH): ax.plot(timeGRD, mtrxPT[i])
32     ax.set_title("BS model Sobol'列でのパス")
33     ax.set_xlabel("年"); ax.set_ylabel("株価")
34     ax.grid(linestyle='--',linewidth=1); ax.set_xticks(timeGRD) ;
```

```
<<mtrxPT>>
 [[100.       101.00502  102.02013  103.04545]
 [100.       109.18608  102.02013  111.39178]
 [100.        93.43694  102.02013   95.3245 ]
 [100.        97.35625   94.78238   99.32297]
 [100.       115.35347  133.06424  117.68377]
 [100.       104.79054   92.67814   90.22795]
 [100.        88.44132   92.67814  106.90745]
 [100.        91.1705    87.03343   83.08409]]
```

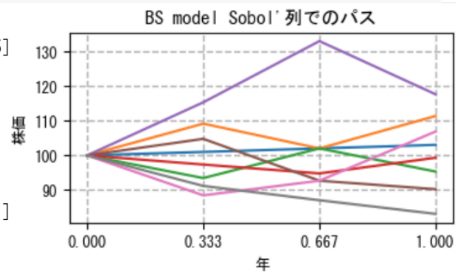


図 6.20: 準乱数 (ソボル列) から価格パスの作成

1〜13： 確率過程のオブジェクト bsmPROC を作成。

- 図 6.18(216 ページ) の 1〜13 行目のコピー。

20〜26： 18 行目で bsmPROC 過程とソボル列から株価を生成する pathGEN オブジェクトが作られ、22 行目の next().value() によって、株価パスを 1 つづつ作成。

- この流れは図 6.13 と全く同じで、図 6.13(201 ページ) の 11〜16 行目のコピー。
- 尚 26 行目でリスト mtrxPT を後続の処理の為 numpy 配列へ変更し、数値確認の為プリントアウト。

30〜34： 23 行目の mtrxPT リストに保存された各パスをグラフ描写。

- ここは図 6.15(図 6.15) の 1〜6 行目のコピー。
- パス数が多くなると、このグラフ描写に時間が掛かる為 29 行目で `if nSTEP<5` の条件を設定。

(ソボル列作成のコード)

ではソボル列に関する 15〜18 行目に移るが、これらの行も図 6.13(201 ページ) の 5〜9 行目と比較しならが説明しよう。

uniRNG： 図 6.13 の 6 行目 uniRNG =UniformRandomGenerator(· · ·) が無い。

- ソボル列は "列" という名前が示すように数列として作成される為、UniformRandomGenerator は無い。

16:uniSeqRNG, 17:gsSeqRNG, 18:pathGEN：

これらの変数は図 6.13 の変数と役割が同じな為、敢えて同じ名前とした。

- 右辺の各コンストラクタは図 6.13 とは異なっている点に注意。

16: `UniformLowDiscrepancySequenceGenerator`：

図 6.13 の UniformRandomSequenceGenerator と同じように準乱数列を作成するコンストラクタ。

- 引数は nSTEP と nSEED。(seed 値はこのコンストラクタで指定)

17: `GaussianLowDiscrepancySequenceGenerator`，
18: `GaussianSobolPathGenerator`：

図 6.13 の GaussianRandomSequenceGenerator 及び GaussianPathGenerator と仕様は全く同じで、若干名前が異なるだけ。

このように準モンテカルロ法は擬似乱数のモンテカルロ法の計算手順 (図 6.13) と殆ど同じとなる。

6.7.3 LSM 手計算

では図 6.19 の MCAmericanEngine が算出した 5.5043 円を再現しよう。長い説明になるので、結論を先に示そう。図 6.23 が LSM 手計算のコードとなり、その出力データが図 6.22 で、最後の行に "NPV: 5.50430" が計算できている。(コードより出力を先に載せた点に注意)

```
1 # ディスカウントファクターDFの準備 stepDF:1期間のDF, colDF:各列に対するDF
2 colDF  = [rfOBJ.discount(jj) for jj in timeGRD[1:]]
3 print('colDF',nA(colDF[:3]))
4
5 # 各時点の行使価値(キャッシュフロー)の用意(マイナス要素をゼロへ)
6 mtrxCF=(mtrxPT-strkPRC)*pC
7 mtrxCF=np.where(mtrxCF<0,0,mtrxCF); print(f'<<mtrxCF{nSTEP}>>¥n',mtrxCF)
```

```
colDF [0.98347 0.96722 0.95123]
<<mtrxCF3>>
 [[-0.        0.        0.        0.      ]
 [-0.        0.        0.        0.      ]
 [-0.        6.56306  0.        4.6755 ]
 [-0.        2.64375  5.21762  0.67703]
 [-0.        0.        0.        0.      ]
 [-0.        0.        7.32186  9.77205]
 [-0.       11.55868  7.32186  0.      ]
 [-0.        8.8295  12.96657 16.91591]]
```

図 6.21: ディスカウントファクター,
キャッシュフローマトリックス (mtrxCF) の準備

では LSM の説明を始める。まず準備として図 6.21 のコードによって、以下の変数を用意する。(これ以降はステップを“時点”とも参照する。例えばステップ 3 は“時点 3”等)

(A. ディスカウントファクターと初期キャッシュフロー行列 matrxCF の準備)

2: colDF： 図 6.20 の 10 行目で作成された 5%フラットのイールドカーブオブジェクト rfOBJ を使い、各時点のディスカウントファクターを colDF に設定。

- colDF[0] は翌期のキャッシュフローを割り引く場合に使用。(図 6.23 の 10 行目参照)

6, 7: mtrxCF： 図 6.20 の 26 行目で numpy 配列に変換した mtrxPT の各要素に対して、6 行目で行使価値を計算し、初期キャッシュフロー行列 mtrxCF (以下 **mtrxCF3** と参照。“3”は時点 3 を意味) を設定。

- 7 行目で where 関数により、マイナスの要素をゼロへ置き換え、プリント。
- mtrxCF は LSM 計算で最も重要な行列で、Longstaff,Schwarts[40] の“Cash-flow matrix”に対応。
- この行列の見方として、mtrx[: , 3](4 列目) は時点 3 の ITM(インザマネー、行使価値) であり、mtrx[: , 2] は時点 2 の ITM 等となる。

LSM のメインの計算は回帰分析で算出する継続価値と比較し、“**mtrxCF のどの要素が行使されるか**”を決定する mtrxCF の編集作業となる。

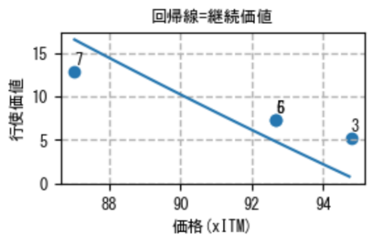
(B. mtrxCF 編集の概観と NPV 計算)

その編集作業をイメージする為、図 6.22 を使い、説明しよう。初めにどの時点で回帰分析による継続価値の計算が必要になるかを確認する。(LSM の計算もツリーと同様、バックワードでの計算)

時点 3： 満期時の為、継続価値はゼロ。(行使価値の計算は必要)
時点 2： 継続価値と行使価値の比較が必要。
時点 1： 継続価値と行使価値の比較が必要。

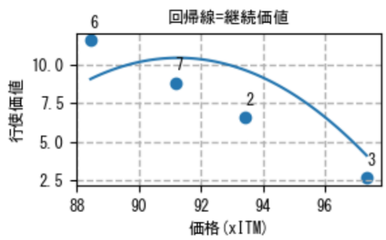
時点 3 の計算は図 6.21 の 6,7 行目で完了しているので、図 6.22 は時点 2, 時点 1 のみで構成されている。中央線の左側が時点 2 で計算されたデータ、右側が時点 1 の計算データを示している。

```
<<時点2 回帰計算概要>>
itmID    [3 5 6 7]
xITM     [94.78238 92.67814 92.67814 87.03343]
行使価値 [ 5.21762  7.32186  7.32186 12.96657]
yITM     [ 0.66584  9.61053  0.       16.63631]
         2
0.01662 x - 5.083 x + 333.1  (時点2 回帰式)
```




```
<<mtrxCF2>>
 [[-0.        0.        0.        0.      ]
 [-0.        0.        0.        0.      ]
 [-0.        6.56306   0.        4.6755  ]
 [-0.        2.64375   5.21762   0.      ]
 [-0.        0.        0.        0.      ]
 [-0.        0.        7.32186   0.      ]
 [-0.       11.55868   7.32186   0.      ]
 [-0.        8.8295    0.       16.91591]]
```

```
<<時点1 回帰計算概要>>
itmID    [2 3 6 7]
xITM     [93.43694 97.35625 88.44132 91.1705 ]
行使価値 [ 6.56306  2.64375 11.55868  8.8295 ]
yITM     [ 4.52222  5.13138  7.20084 16.36134]
         2
-0.1715 x + 31.31 x - 1418  (時点1 回帰式)
```




```
<<mtrxCF1>>
 [[-0.        0.        0.        0.      ]
 [-0.        0.        0.        0.      ]
 [-0.        0.        0.        4.6755  ]
 [-0.        0.        5.21762   0.      ]
 [-0.        0.        0.        0.      ]
 [-0.        0.        7.32186   0.      ]
 [-0.       11.55868   0.        0.      ]
 [-0.        0.        0.       16.91591]]
NPV: 5.50430
```

図 6.22: LSM 手計算の結果 (図 6.23 コードからの出力)

時点 2(左側)： 図 6.21 の mtrxCF3(時点 3) に対して、回帰計算が行われ、左下 mtrxCF2(時点 2) へ編集されている。

- mtrxCF3[:, 3] と mtrxCF2[:, 3](4 列目どうし) を比較すると、mtrxCF2[:, 3] のほうが明らかにゼロ多くなっている。
- 時点 2 において、“ 行使価値 > 継続価値 ” となった為、時点 3 の行使価値がゼロに置換えられている。
- この編集がどのように行われたかは後程詳しく説明する。

時点 1(右側)： 同様に mtrxCF2(時点 2) に対して、回帰計算が行われ、右下 mtrxCF1(時点 1) へ編集されている。

- mtrxCF2[:, 2] と mtrxCF1[:, 2](3 列目どうし) を比較すると、mtrxCF2[6, 2] がゼロに置換えられている。
- この mtrxCF1 は最終的に必要な mtrxCF であり、**行使する時点と行使価値の一覧**が表示されている。
- 各行にゼロ以外の数字は 1 つだけとなり、数字が位置する列が行使する時点、数値が行使価値を示している。

図 6.23 の 43 行目でこの mtrxCF1(コード上は mtrxCF) に図 6.21 で準備したディスカウントファクターリスト colDF を掛けて、NPV5.5043 円が算出される。

図 6.23 の 43: `np.sum(mtrxCF[: ,1:]*colDF)/nPATH`

- スライス `[:,1:]` は mtrxCF の 0 列目 (時点 0) が不要な為、スライスで除外。
- `mtrxCF[:,1:]*colDF` によって、mtrxCF の各列にそれぞれの時点のディスカウントファクターが掛けられ、各要素は現在価値に割り引かれる。
- `np.sum` で全ての要素を合計し、nPATH で割ることで NPV を得る。

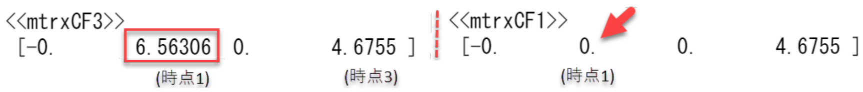
(C. mtrxCF の編集方法の具体例)

では LSM の mtrxCF 編集方法を説明する。図 6.20 の mtrxPT(株価のパス行列) を含め、mtrxCF3 から mtrxCF1 への変化を代表的なパスで例示しよう。(mtrxPT, mtrxCF は上の行からパス 0, · · · , パス 7 として参照)

パス 0,1： 図 6.20 の mtrxPT が本来の株価推移。

- 行使価格 100 円の Put の為 パス 0, 1 に ITM となる株価は無く、mtrxCF のパス 0, 1 の要素は全てゼロ。

パス 2： mtrxCF3(図 6.21) のパス 2 は時点 1 と 3 で ITM。



- 時点 1 までバックワードが進むと、mtrxCF1 の時点 1 の行使価値はゼロに編集。その結果 mtrxCF1 は時点 3 で行使が行われるパスを示している。

- 時点 1 で計算された継続価値が行使価値 6.56306 を上回っていた為、行使価値はゼロへ。

パス 6： mtrxCF3 のパス 6 は時点 1 と 2 で ITM。

```
<<mtrxCF3>>                                    <<mtrxCF1>>
 [-0.      11.55868  7.32186  0.      ]        [-0.      11.55868  0.        0.      ]
           (時点1)   (時点2)                                       (時点2)
```

- 時点 1 までバックワードが進むと、時点 1 の行使価値 11.55868 は継続価値より大きかった為、時点 1 で行使となる。従って 時点 2 の行使価値はゼロへ修正。
- この結果 mtrxCF1 は時点 1 で行使が行われるパスとなる。

繰り返しとなるが、mtrxCF3 では 1 つのパスに複数の行使価値が表示されている。一方、LSM の編集後、mtrxCF1 は 1 つのパスに行使価値は 1 回または 0 回となる。(mtrxCF1 の行使価値を“1”に替えれば Longstaff,Schwarts[40] の“Stopping rule”表に対応。)

(D. 説明変数 x, 従属変数 y の準備)

LSM では**回帰分析** (**regression analysis**) で継続価値を計算するが、この回帰分析は次の変数 x、y で行われることを覚えよう。

説明変数 x： 当期に ITM となっている価格。

従属変数 y： 説明変数 x と同じパスにある翌期以降の (割引後) 行使価値。(この説明から判るように従属変数 y を準備することは多少の手間が必要)

ここでは図 6.22 の右列“時点 1 回帰計算概要”の下に表示された 4 項目とデータを使い、回帰分析の x、y の設定方法を数値例で説明する。(数値の出所を各自確認)

itmID[2 3 ···]: mtrxPT(図 6.20) の時点 1(2 列目) で ITM となったパス番号。(図 6.23 コード 6 行目で設定)

xITM[98.43694 ···]: itmID の株価のリスト。この xITM が説明変数。(図 6.23 コード 13 行目で設定)

xEXE(行使価値)[6.56306 ···]: xITM の行使価値のリスト。コードの中では変数名“xEXE”とした。(図 6.23 コード 5 行目で設定)

- xITM は時点 1 での行使価値であり、従属変数ではない点に注意。

yITM[4.52222 ···]: yITM が従属変数。(図 6.23 の 13 行目で設定)

- “従属変数 y の準備”とは、回帰計算前に行う作業であり、時点 1 の直前に編集された mtrxCF2 (図 6.22 の左列) で行使価値を探すことが必要。
- yITM は mtrxCF2 の各 itmID 上にある行使価値で構成。
- yITM の数値が mtrxCF2 より若干小さい理由は時点 1 へ割り引かれている為。

  yITM[2]=7.20084 : この値は mtrxCF2[6,2]=7.32186 に colDF[0] =0.98347 を掛けた値。

  yITM[3]=16.36134 : この値も mtrxCF3[7,3]=16.91591 であるが、時点 3 の行使の為、2 期分ディスカウント (colDF[0]**2) された値。(図 6.23 コード 10 行目でディスカウント処理)

(D 部分を処理するコード)

説明変数と従属変数が準備できれば、Python による回帰分析は非常にシンプルなコードで完了する。図 6.23 において、D 部分を処理するコードは 16 行目までとなる。以下、簡単な説明を記しておく。

2: for ss in range(nSTEP-1, 0, -1) : 上記 B で確認したように、バックワードで時点 2, 時点 1 の順に計算。ss 変数が時点を表す。

4,5: `copy` : mtrxPT と mtrxCF から ss 列を抜き出し、時点 ss の株価 xPRC と行使価値 xEXE を作成。

- 配列のメソッド copy でいわゆる**値複写**を行っている。(copy で作成しない場合、mtrxCF の修正が xEXE を変える)

6: itmID=np.`where`(xEXE>0) : where 関数を使い、xEXE の要素の中で ITM (インザマネー) となっている要素位置 (座標) を itmID に設定。

- itmID を**ブールインデックス** (`Boolean index`) と言う。

10: `yEXE += mtrxCF[:, ii+1].copy()*colDF[0]**(ii+1-ss)` :

従属変数の元となる yEXE を作成。

- 13 行目で yITM = yEXE[itmID] によって、この yEXE から ITM のみを取り出し、従属変数 yITM を作成。
- D で説明した次の条件がコーディングされている。
  - ◇ 時点 ii の直前に編集された mtrxCF(ii+1) で行使価値を検索 => mtrxCF[:,ii+1].copy()

```
 1 # mtrxCFの編集
 2 for ss in range(nSTEP-1, 0, -1):     # 最後から2列目のスタートでバックワード
 3     # 説明変数 : xPRC(otmIDを除く)
 4     xPRC  = mtrxPT[:,ss].copy()      # ss時点 株価
 5     xEXE  = mtrxCF[:,ss].copy()      # ss時点 行使価値
 6     itmID = np.where(xEXE>0)         # ss時点 itmの位置
 7     # 従属変数 : yEXE = ss時点より先の行使価値を合計(価格はss時点へ割引)
 8     yEXE  = np.zeros(nPATH)
 9     for ii in range(ss,nSTEP):
10         yEXE += mtrxCF[:,ii+1].copy()*colDF[0]**(ii+1-ss)
11 
12     # 回帰分析 (ITMのみ)                           xEXEmはグラフ用
13     xITM  = xPRC[itmID] ; yITM = yEXE[itmID] ; xEXEm = xEXE[itmID]
14     coef  = np.polyfit(xITM, yITM, 2)
15     # ss時点 継続価値(マイナスの継続価値は0, contVL=continueValueの略)
16     contVL = np.poly1d(coef)(xPRC); contVL = np.where(contVL<0, 0, contVL)
17 
18     # 行使パスの特定とmtrxCFの編集 (継続パスはチルダ~を付けて参照)
19     exeID  = xEXE > contVL        # ">"はnumpy論理演算子
20     mtrxCF[:,ss+1:][exeID] = 0    # ss列で行使, ss+1列以降の要素はゼロ
21     mtrxCF[:,ss][~exeID]   = 0    # ss列で行使されない要素はゼロ
22 
23     # 回帰式とmtrxCFの出力 (nSTEP<5でプリント)
24     if nSTEP<5:
25         # 回帰式
26         print(f' <<時点{ss} 回帰計算概要>>')
27         print('itmID   ',itmID[0]) ; print('xITM   ', xITM)
28         print('行使価値' ,xEXEm)      ; print('yITM   ' ,yITM);
29         print(np.poly1d(coef), f' (時点{ss} 回帰式)')
30         # 回帰グラフ
31         xRNG  = np.arange(xITM.min(),xITM.max(),0.1)     # x軸レンジ
32         cLine = np.poly1d(coef)(xRNG)                    # 継続ライン
33         _, ax = plt.subplots(figsize=(3, 1.5))
34         ax.scatter(xITM, xEXEm)     ; ax.plot(xRNG, cLine)
35         ax.set_xlabel("価格(xITM)"); ax.set_title("回帰線=継続価値",fontsize=9)
36         ax.set_ylabel("行使価値")    ; ax.grid(linestyle='--',linewidth=1)
37         for ii,_ in enumerate(xITM):                     # 番号表示
38             ax.text(xITM[ii], xEXEm[ii]+0.8, itmID[0][ii])
39         # グラフとmtrxCFの表示
40         plt.show(); print(f'<<mtrxCF{ss}>>¥n', mtrxCF);print('¥n')
41 
42 # American put NPV(ここでcolDFを使用)
43 print(f'NPV: {np.sum(mtrxCF[:,1:]*colDF)/nPATH:.5f}')  # mtrxCFの1列目は無視
```

図 6.23: LSM の手計算コード (図 6.22 を出力)

◇ 時点 ii 時点へ割引き => colDF[0]**(ii+1-ss)

(**演算子はべき乗)

13: xITM, yITM： 説明変数 xITM と従属変数 yITM を作成。

- 6 行目で作成したブールインデックス itmID を使い、**xPRC[itmID]** とすること (**ブールインデックス参照**と言う) でインザマネーのみの株価の配列を作成。
- yITM, xEXEm(グラフ表示用) も同じ処理。

14: coef = np.polyfit(xITM, yITM, 2)：

polyfit 関数で最小二乗法の係数を算出し、係数オブジェクト coef に保存。

- polyfit 関数の引数は (説明変数 x, 従属変数 y, 回帰式の次数)。
- 回帰式の次数は式 (6.48) では 2 次。

16: contVL = np.poly1d(coef)(xPRC) :

poly1d 関数は初めのカッコで係数オブジェクト coef を、2 番目のカッコで説明変数データ等を与えて、**最小二乗推定値**を算出。

- ここでは 4 行目で設定した xPRC(ss 時点での株価) を与え、全てのパスの継続価値を算出し、contVL 変数 (continuing value の略) へ保存後、マイナス値をゼロへ修正。
- poly1d 関数で 2 番目のカッコを無くした **poly1d(coef)** とすると、29 行目のように回帰式を出力。

(polyfit, poly1d への補足)

- polyfit, poly1d は対で使用することが多い。
- この 2 つの関数には新しいものが用意され、np.polyfit が np.polynomial.polynomial.polyfi poly1d が np.polynomial.polynomial.polyval。[*24]
- 新しい polyfit では係数が逆に出力され、式 (6.48) の係数では [c, b, a] の順。

(E. 継続価値計算とキャッシュフロー行列 mtrxCF の編集)

図 6.22 “ 時点 1 回帰計算概要 ” 5 行目には式 (6.48) に対応した回帰式

$$-0.1715x^2 + 31.31x - 1418 \qquad \text{(時点 1 回帰式)}$$

を表示させた。

その下は x 軸を価格、y 軸を行使価値としたグラフで、xITM と行使価値をドットでプロットし、パス番号を振った。回帰式から計算された継続価値を実線で記入しているが、この実線より上のドットでは行使が起こることになる。

この 4 つのドットがどのように mtrxCF2 を修正し、mtrxCF1 となるかを見て行こう。

パス 2, 3, 7 : この 3 つのドット (行使価値) は回帰線 (継続価値) より下に位置しているので、行使されない。mtrxCF2 時点 1 の 6.656306, 2.64375, 8.8295 は mtrxCF1 でゼロに置換え。

[*24] polyval 関数では回帰式を書く “ poly1d(coef) ” 機能は提供されていないと思われるが、回帰係数が判れば 問題は無いだろう。また見て判る通り 関数を呼び出すコードは非常に長く、np.polynomial.polynomial ライブラリを別途 import したほうが良い。

```
  <<mtrxCF2>>                                  <<mtrxCF1>>
2 [-0.        6.56306  0.         4.6755 ]     [-0.        0.         0.         4.6755 ]
3 [-0.        2.64375  5.21762  0.       ]     [-0.        0.         5.21762  0.       ]
7 [-0.        8.8295   0.        16.91591]     [-0.        0.         0.        16.91591]
             (時点1)                                       (時点1)
```

パス 6： 回帰線より上なので、時点 1 で行使。mtrxCF2 時点 2 にある 7.32186 は mtrxCF1 でゼロに置換え。

```
  <<mtrxCF2>>                                    <<mtrxCF1>>
6 [-0.        11.55868  7.32186  0.        ]    [-0.        11.55868  0.         0.        ]
                       (時点2)                                        (時点2)
```

(E 部分のコード)

最小二乗法計算の後の継続価値は図 6.23 の 16 行目で計算され、contVL 変数に設定されている。(この 16 行目は“D 部分を処理するコード”の最後で説明済み)

キャッシュフロー行列 mtrxCF を編集するコードは 19, 20, 21 行の 3 行のみで、上で説明した“パス 2,3,7”と“パス 6”の処理をそのまま 記述している。

19: exeID = xEXE > contVL :

- この行は“exeID = ブールインデックス”を意味。
- 右辺 `xEXE > contVL` が“行使価値 > 継続価値”となるブールインデックスであり、左辺 exeID に設定。

20: mtrxCF[: , ss+1:][exeID]= 0 :

- 前半の mtrxCF[: , ss+1:] は mtrxCF 行列の ss+1 時点とそれより後の時点の列を指定。
- 後半の [exeID] はブールインデックス参照で、ss 時点で行使されるパスの ss+1 時点以降の要素をゼロに置換え。(パス 6 で行った処理)

21: mtrxCF[: , ss][~exeID]= 0 :

- 前半の mtrxCF[: , ss] は mtrxCF 行列の ss 列を指定。
- exeID の前にある **~(チルダ)** はブールインデックスの真偽を反転させる演算子。
- exeID は行使されるパス番号が指定されているので、~exeID は ss 時点で継続するパス番号が指定されている。
- ブールインデックス参照 [~exeID] によって ss 時点の継続するパスをゼロへ置換え。(パス 2,3,7 で行った処理)

24〜40 : これらの行は計算結果を出力する為のコードで、次の“ 図 6.22 を出力するコード ”で説明。

43: np.sum(mtrxCF[: ,1:]*colDF)/nPATH

- 43 行目でアメリカンオプションの NPV を計算しているが、この行は“ B. ···NPV 計算 ”の最後で説明済み。

(図 6.22 を出力するコード)

24〜40 行までは図 6.22 を出力する為のコードで LSM との関連は薄い。以下、注意すべき部分を簡単にメモしておく。

31: xRNG = np.arange(xITM.min(),xITM.max(),0.1) :

- Python の range 関数は刻み幅が整数に限定。この numpy の arange 関数は刻み幅に実数の指定が可能。
- numpy 配列の最大値/最小値は max/min メソッドで取得。

32: np.poly1d(coef)(xRNG) : 14 行目で計算した coef と xRNG から継続価値ラインのデータを計算。

37, 38: for ii,_ in enumerate(xITM):
ax.text(xITM[ii], xEXEm[ii]+0.8, itmID[0][ii]+1)

この for ループによって、各ドットの上にパス番号を表示。

- 37 行目 for ii,_ in enumerate(xITM) はプロットに使った xITM リストに対し、`enumerate` 関数により、リストのインデックス番号のみを取得。
- 38 行目 ax オブジェクトの text メソッドは **ax.text(x 座標，y 座標，テキスト)** によって、座標の位置にテキストを表示。
- インデックス番号 ii を使うと 各ドットの x 座標は xITM[ii], y 座標は xEXEm[ii]。
- パス番号は itmID[0][ii] であり、text メソッドの 3 番目の引数に指定。
- y 座標の+0.8 はパス番号とドットが重ならない様、上に+0.8 ずらした。
- 図にテキストを書きたい場合、このようなコードとなる。

6.8 (補足 1) 正規分布と確率微分方程式 *

この補足節では正規分布と確率微分方程式の関係と伊藤の公式について説明する。これらのテーマに関して、学習済みの読者は読み飛ばそう。(この節は“ 厳密さ ”より“ 判り易さ ”に重きを置いて記述した為、正確でない表現がある点に注意)

6.8.1 正規分布の変数変換

初めに正規分布について簡単にまとめておく。

平均 0、分散 1 の正規分布 $N(0,1)$ を**標準正規分布**と呼び、水平軸とで囲まれる面積は 1.0 となる。

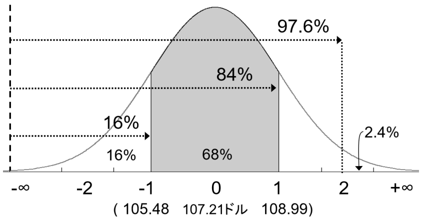


図 6.24: 標準正規分布

標準正規分布の累積密度関数 $N(d)$ とはマイナス無限大 $(-\infty)$ から変数 d の位置までの面積をあらわす関数である。[*25]

例えば、d が -1 の場合 $N(-1) \simeq 0.16$ という面積となる。同様に $d = 0$ では $N(0) = 0.50$、$d = 1$ で $N(1) \simeq 0.84$ であり、この面積が d の位置の確率を表わしている。式 (6.4) や式 (6.5) ではこの $N(d)$ が使用され、6.2.3 節で先物価格の分布に確率を与えた。

正規分布は左右対称な為、図 6.4 で先物価格が 1 標準偏差の価格 108.99 ドル以上となる確率は $1 - N(1) =$ 約 0.16 とするのでは無く、d の符号を逆転させ、$N(-1) =$ 約 0.16 を計算する点に注意しよう。

平均 μ、標準偏差 σ の正規分布に従う確率変数 $Y \sim N(\mu, \sigma^2)$ は標準化への**変数変換**

$$z = \frac{y - \overbrace{\mu}^{\substack{\text{平均} \\ \text{(期待値)}}}}{\underbrace{\sigma}_{\text{標準偏差}}} \tag{6.49}$$

によって 標準正規分布に従う確率変数 $Z \sim N(0,1)$ に変換後、上で述べた標準正規分布の確率を当てはめる作業や数式展開を行う。(通常、大文字 Y、Z は確率変数を表し、小文字 y、z はその実現値を表す)

[*25] 正規分布の記号 N(平均, 分散) と累積密度関数 $N(d)$ はいずれも $N()$ が使われるが、前者は引数が 2 つ、後者は 1 つとなる点で区別。

6.8.2 d_2 の解釈

式 (6.6) d_2 の計算式はこの変数変換後の数式となっている。まず d_2 の符号は反転しているので、マイナス 1 を掛けて、符号を元に戻そう。 現在の先物価格を F_0 とすると、式 (6.6) を次式となる。

$$-d_2 \quad = \quad \frac{\ln X - \ln F_0}{\sigma\sqrt{T}} + \overbrace{\frac{\sigma\sqrt{T}}{2}}^{\substack{\text{伊藤の公式から}\\ \text{計算される項}}} \tag{6.50}$$

分子の F_0や X は対数正規分布を仮定している為、対数を取ることで正規分布の変数に変えている。右辺第 1 項と標準化の変数変換の式 (6.49) を比較すると同じ形となっていることが判るだろう。

以下この節の残りは式 (6.50) 第 2 項の説明であり、伊藤の公式と正規分布の式変形となる。

価格 F_t の分布の前提は式 (6.15) であったが、σ は 1 年当たりの標準偏差 (ボラティリティ) なので、分散の項に時間 t が掛けられた次式となる。

$$\text{価格収益率} : \frac{dF_t}{F_t} \quad \sim \quad N(\underbrace{0}_{\text{平均}}, \underbrace{\sigma^2 \ t}_{\text{分散}}) \tag{6.51}$$

今日を起点とし、将来の時点 T までの先物 F_T の収益率の分布も正規分布である。この計算には伊藤の公式 (6.8.4 節で説明) が利用され、計算結果は次のような正規分布となる。

$$\text{時点 } T \text{ の収益率分布} : \ln\frac{F_T}{F_0} \quad \sim \quad N\Big(\underbrace{-\frac{1}{2}\sigma^2 \ T}_{\text{平均}}, \underbrace{\sigma^2 \ T}_{\text{分散}}\Big) \tag{6.52}$$

左辺分母の $\ln F_0$ を右辺に移して、

$$\text{対数価格の分布} : \ln F_T \quad \sim \quad N\Big(\underbrace{\ln F_0 - \frac{1}{2}\sigma^2 \ T}_{\text{平均}}, \underbrace{\sigma^2 \ T}_{\text{分散}}\Big) \tag{6.53}$$

を得る。つまり、オプション満期時点の価格 F_T の対数を取った値は上式 (6.53) の平均と分散で正規分布している。この正規分布を式 (6.49) で標準化させると、

$$z \quad = \quad \frac{y-\mu}{\sigma} \quad = \quad \frac{\overset{(\ln X)}{y} - \overbrace{\Big(\ln F_0 - \frac{1}{2}\sigma^2 \ T\Big)}^{\text{平均}}}{\underbrace{\sigma\sqrt{T}}_{\text{標準偏差}}} \quad = \quad \frac{\overset{(\ln X)}{y} - \ln F_0}{\sigma\sqrt{T}} + \frac{\sigma\sqrt{T}}{2}$$

となる。ここで $y = \ln X$ を代入すれば、上式 (6.50) を得る。更に z の符号を反転させた $-z$ が式 (6.6) の d_2 である。

6.8.3 満期時点の価格の計算式

ブラックモデルや BS モデルを表す数式に必ず記入されている項が W や dW であるが、これは**ブラウン運動** (Norbert Wiener 博士の頭文字 W) を意味する。時点 T のブラウン運動 $W(T)$ の位置を正規分布で記述すると 次式となる。($W(t) = W_t$ で、ここでは時間を強調する為、左辺の表記とした)

$$W(T) \quad \sim \quad N(0, \underbrace{T}_{\text{分散}}) \; = \; \sqrt{T} \times \underbrace{N(0,1)}_{\text{標準正規分布}} \tag{6.54}$$

つまり ブラウン運動の時点 T の位置は分散 T の正規分布で記述される。右辺は分散 T を $N(\cdots)$ の外へ出している。標準正規分布に従う変数を $z \sim N(0,1)$ とすれば、

$$W(T) \quad = \quad z \times \sqrt{T} \tag{6.55}$$

と表される。ブラウン運動の微小時間 dt の変動幅は

$$dW(t) \quad \sim \quad N(0, dt) \; = \; \sqrt{dt} \times N(0,1) \tag{6.56}$$

であり、時点 0 から T までの積分は次式で計算される。

$$\int_0^T dW(t) \quad = \quad \Big[W(t) \Big]_0^T \; = \; W(T)\;, \qquad \because W(0) = 0 \tag{6.57}$$

ここまでは特別な計算は行っていない。取り敢えず、ブラウン運動の記号の操作に慣れよう。

次に 式 (6.51) では " 証券価格の収益率が正規分布する " ことをボラティリティσ を使って

$$\text{価格収益率} : \frac{dF_t}{F_t} \quad \sim \quad N\big(\; 0,\; \sigma^2 \; t \;\big)$$

と表したが、dW_t を持つ確率微分方程式としては次式で表される。(正規分布と確率過程を相互に書き換えよう。すると確率微分方程式の意味が判りやすくなる)

$$\frac{dF_t}{F_t} \quad = \quad 0 \; dt + \sigma \; dW_t \tag{6.58}$$

この方程式は有名な伊藤の公式 (6.66) を経由して、

$$d\Big(\ln F_t\Big) \quad = \quad 0 - \frac{1}{2}\sigma^2 \; dt + \sigma \; dW_t \tag{6.59}$$

と変形出来、両辺を時点 0 から T まで積分すると、左辺は

$$\int_0^T d\Big(\ln F_t\Big) \quad = \quad \Big[\ln F_t\Big]_0^T \; = \; \ln F_T - \ln F_0 \; = \; \ln \frac{F_T}{F_0}$$

なので、次式を得る。

$$\ln \frac{F_T}{F_0} \quad = \quad 0 - \frac{1}{2}\sigma^2 \; T + \sigma \; W(T) \; = \; -\frac{1}{2}\sigma^2 \; T + \sigma\sqrt{T} \; z \tag{6.60}$$

上式は式 (6.52) と同じ記述となっていることを確認しよう。この式から対数 ln を取り除くと、

$$F_T \quad = \quad F_0 \exp\Big[\frac{1}{2}\sigma^2\ T + \sigma\ z\sqrt{T}\Big] \tag{6.61}$$

F_0 : 現在の先物価格、　　F_T : T 時点の先物価格

を得る。[*26] つまり 現在の価格 F_0 とボラティリティσ があれば、将来の T 時点の価格 F_T を計算できる式であり、“ 確率微分方程式 (6.58) の解 ” である。(” 解 ” に関しては 6.8.4 節で説明)

6.8.4　伊藤の公式と確率微分方程式の解

ブラックショールズモデルで有名となった “ 伊藤の公式 ” がどんな場合に必要となるかを理解しよう。例えば 第 7 章の “ ノーマルモデル ” では伊藤の公式を必要としない。つまり 対数正規分布に従う確率変数を処理する場合に必要となることが予想できる。

式 (6.58) の確率微分方程式から式 (6.59) を得るには伊藤の公式が必要であり、式 (6.59) 以降は通常の式変形である。初めに式 (6.58) のように左辺が分数になっていない式 (6.59) のようなタイプ、

$$dF_t \quad = \quad a\ dt + \sigma\ dW_t \tag{6.62}$$

の方程式について考えよう。このタイプが第 7 章 “ ノーマルモデル ” の確率微分方程式である。まず このタイプを解くことを考える。

この方程式を “ 解く ” とは “ 時点 T において、F_T の値がどうなっているかを計算できる式 ” を得ること意味するが、その為に両辺を 0 から T まで積分し、次式を得る。

$$\int_0^T dF_t \quad = \quad a\int_0^T dt + \sigma\int_0^T dW_t \tag{6.63}$$

$$\Big[\ F_t\ \Big]_0^T \quad = \quad a\Big[\ t\ \Big]_0^T + \sigma\Big[\ W(t)\ \Big]_0^T$$

式 (6.62) の解として、

$$F_T \quad = \quad F_0 + aT + \sigma\ W(T) \tag{6.64}$$

のように F_T を現時点の値である F_0 と a, σ 及び T 時点のブラウン運動 $W(T)$ で表すことが出来ている。

[*26] この部分の数式展開は多くのテキスト、例えば、西村 [7]　91 ページ等に記載されているので、本書では省略する。d_2 の導出も同 [7]　104 ページを参考にした。

しかし 式 (6.58) の方程式では 左辺分母の F_t を右辺に移行させ、

$$dF_t = aF_t\ dt + \sigma F_t\ dW_t \qquad (\text{ただし } a = 0) \tag{6.65}$$

となるが、上で説明したように単純に積分しても、F_T を F_0 で表すことが出来ない。ここで必要となるのが伊藤の公式である。[*27]

伊藤の公式

F_tの確率過程を $dF_t = aF_t\ dt + \sigma F_t\ dW_t$ 、x と時間 t の非確率的関数を $g(t,x)$ として、$x = F_t$とした場合の関数 g は

$$dg(t,F_t) = \frac{\partial g}{\partial t}(t,F_t)dt + \frac{\partial g}{\partial x}(t,F_t)dF_t + \frac{1}{2}\frac{\partial^2 g}{\partial x^2}(t,F_t)\big(dF_t\big)^2 \tag{6.66}$$

という確率過程に従う。

ただし $\dfrac{\partial g}{\partial t}(t,F_t) \ := \ \dfrac{\partial g(t,x)}{\partial t}\Big|_{x=F_t}$ 等であり、$(dF_t)^2$の計算では、

$$dt{\times}dt = 0\ ,\ dt{\times}dW_t = 0\ ,\ dW_t{\times}dW_t = dt \tag{6.67}$$

を使用する。

対数正規分布する確率変数に対しては、関数 $g(t,x) = \ln(x)$ のように設定することが常套手段で、新しい確率過程の係数は次式で算出される。

$$\frac{\partial g}{\partial t} = 0、\qquad \frac{\partial g}{\partial x} = \frac{1}{x}、\qquad \frac{\partial^2 g}{\partial x^2} = -\frac{1}{x^2}$$

$a = 0$ で $dF_t = \sigma F_t\ dW_t$ であり、

$$\begin{aligned} dg\big(t,F_t\big) &= d\big(\ln F_t\big) \ = \ 0 + \frac{1}{F_t}dF_t + \frac{1}{2}\Big(-\frac{1}{F_t^2}\Big)\big(dF_t\big)^2 \\ &= \frac{1}{F_t}\big(\sigma F_t\ dW_t\big) + \frac{1}{2}\Big(-\frac{1}{F_t^2}\Big)\big(\sigma F_t\ dW_t\big)^2 \\ &= \sigma dW_t - \frac{1}{2}\sigma^2 dt \qquad (\text{式 6.67 を使用}) \end{aligned}$$

が新しい確率過程となり、式 (6.59) を得る。この式に対して、式 (6.63) のように両辺を 0 から T まで積分すると、

$$\begin{aligned} \int_0^T d\Big(\ln F_s\Big) &= -\frac{1}{2}\sigma^2 \int_0^T dt + \sigma \int_0^T dW_t \qquad \text{より、} \\ \Big[\ \ln F_t\ \Big]_0^T &= \Big[\ -\frac{1}{2}\sigma^2 t\ \Big]_0^T + \Big[\ \sigma\ W_t\ \Big]_0^T \end{aligned}$$

であり、

$$\ln F_T - \ln F_0 = \ln\frac{F_T}{F_0} = -\frac{1}{2}\sigma^2 T + \sigma\ W(T)$$

と式 (6.60) を得る。

[*27] 式中に現れる $\partial g(t,x)/\partial t\big|_{x=F_t}$ は t で $g(t,x)$ を偏微分後、F_t を x へ代入することを意味。尚、この公式の記述は村上 [9] 22〜24 ページを参照した。

繰り返しになるが、伊藤の公式とは非確率的関数 $g(t,x)$ の x に確率過程 F_t を代入した場合、関数 g がどのように記述されるかを導く公式である。関数 g を上手く設定することで確率微分方程式を解くツールとなる。

(伊藤の積公式と商公式)

6.5.2 節で示したようにモデル式を変形する場合、価格をニューメレールで掛けたり、割ったりして、新しい確率過程を作り出す必要が出てくる。その場合に役立つ公式が伊藤の積公式, 伊藤の商公式である。(公式のみを記す。式の導出は村上 [9] を参照。)

伊藤の積公式と商公式

F_1, F_2が次の確率過程 (相関ρ) に従う時

$$\frac{dF_1}{F_1} = \mu_1 dt + \sigma_1 dW_1, \quad \frac{dF_2}{F_2} = \mu_2 dt + \sigma_2 dW_2, \quad dW_1 dW_2 = \rho dt$$

F_1, F_2の積と商の確率過程は次式で表される。

$$\text{(積公式)}\ \frac{d(F_1F_2)}{F_1F_2} = \big(\mu_1 + \mu_2 + \rho\sigma_1\sigma_2 \qquad \big)dt + \sigma_1 dW_1 + \sigma_2 dW_2 \tag{6.68}$$

$$\text{(商公式)}\ \frac{d(F_1/F_2)}{F_1/F_2} = \big(\mu_1 - \mu_2 - \rho\sigma_1\sigma_2 + \sigma_2^2\big)dt + \sigma_1 dW_1 - \sigma_2 dW_2 \tag{6.69}$$

第7章　Bachelier ノーマルモデルとクラス作成

この章では金利の分布を正規分布と仮定するノーマルモデルの概要を説明し、その例としてキャップ/フロアー、スワップションの計算を紹介する。
ノーマルモデルは別名 Bachelier モデル (バシェリエと読む) とも言われ、Louis Bachelier が 1900 年に発表した”Theory of speculation in The Random Character of Stock Market Prices”が起源となる。金利がプラスの時代はほとんど使用されなかったモデルであったが、マイナス金利の発生以降、標準的モデルの 1 つとなった。

7.1　金利先物の概要

3ヶ月物金利先物とそのオプションについて説明しよう。円建て 3ヶ月物預金金利の先物としては、東京金融取引所に上場している**ユーロ円 3ヶ月金利先物** (以下 Tibor 先物) [*1] があり、ユーロ建て預金では **3ヶ月 Euribor 先物** (以下 Euribor 先物) が ICE Futures Europe で取引されている。

US ドルでは Libor の公表停止により、**3ヶ月ユーロドル先物**の後継商品である **3ヶ月 SOFR 先物** (以下 SOFR 先物) がシカゴ・マーカンタイル取引所 (CME) に上場され、活発な取引が行われている。[*2]

これらの金利先物はユーロドル先物の商品設計がベースとなり、Tibor 先物や Euribor 先物はその類似商品である。

7.1.1　3ヶ月ユーロドル先物

初めにユーロドル先物 (以下 ED 先物) の取引イメージを図 7.1 で説明しよう。ED 先物は上場廃止となっているが、Tibor 先物や Euribor 先物は ED 先物と同様の商品設計であり、また SOFR 先物を理解する為にも、ED 先物の商品知識は必要である。

[*1] Tibor には国内 Tibor と**ユーロ Tibor** の 2 種類がある。後者は午前 11 時時点の東京オフショア市場での銀行間貸し出し金利として日本の全国銀行協会 (JBA) が公表する金利である。尚 1.3.1 節で説明した Tibor は国内 Tibor である。
[*2] 1ヶ月 SOFR 先物も CME に上場されているが、本書では扱わない。

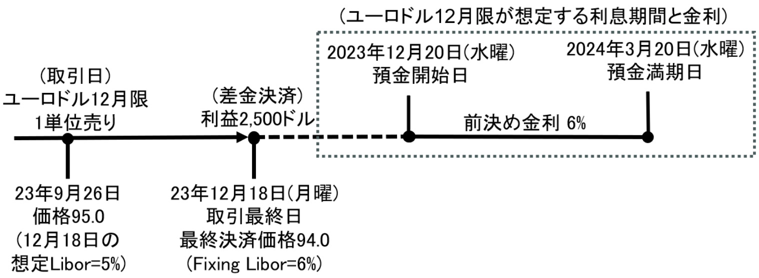


図 7.1: ユーロドル先物のイメージ

限月は債券先物同様、主に 3、6、9、12 月であり、**各限月の第 3 水曜日**に始まる 3ヶ月物預金金利がこの先物の原資産であった。

また Libor の慣習から、その金利が決定される日は 2 営業日前であり、第 3 水曜の 2 営業日前が先物の取引最終日となっていた。

先物価格は 100 から金利を引いた値で建値され、この **100− 金利**を **IMM 指数** (**IMM Index**) と言う。[*3] 例えば図 7.1 取引日 (9 月 26 日、左下) の 12 月限先物価格が 95.0 であった場合、" 取引最終日 (12 月 18 日) に決定される Libor レートは 5.0%(= 100 − 95.0) である " と市場は予想し、先物が売買されている。

取引最終日には、その日に決定された 3ヶ月 Libor レートを使い、最終決済価格が算出される。もし 3ヶ月 Libor が 6.0%となった場合、最終決済価格は 100 − 6.0 = 94.0 (左下 2 列目) となる。

図 7.1 では " 金利が上昇すると予想する投資家が 95.0 で先物を 1 枚 (=**100 万ドル**) 売り立て、最終決済価格 94.0 となった " ことを想定している。 この場合、価格の差は 95.0 − 94.0 =**1 ポイント**であり、IMM 指数の 1 ポイントは 2,500 ドルである為、この投資家の利益は 2,500 ドルと計算される。[*4]

尚、債券先物と異なり、金利先物は**差金決済**が行われるので、この投資家は預金額の 100 万ドルを準備する必要は無い。

7.1.2 3ヶ月 SOFR 先物

図 7.2 は SOFR 先物の 12 月限のイメージ図である。SOFR 先物はユーロドル先物の後継商品であり、次の 3 点は SOFR 先物に引き継がれている。

- IMM 指数 (100 − 金利) での建値

[*3] " IMM " に関しては、383 ページの脚注を参照。

[*4] " IMM 指数の 1 ポイントを 2,500 ドルとする " 点に関しては、売買している対象が 3ヶ月預金の " 金利 " の為。この投資家の利益額は 5.0%の 3ヶ月預金の利息額と 6.0%の利息額の差の 1.0%利息額であり、100 万ドル × 1.0% × 3ヶ月/1 年 = 2, 500 ドル となる。

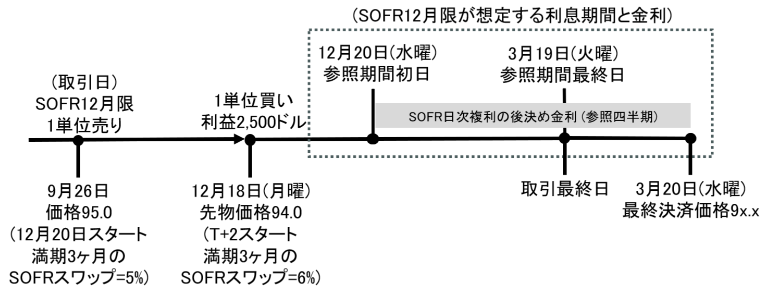


図 7.2: 3ヶ月 SOFR 先物のイメージ

- 取引単位 1 枚 = 100 万ドル
- IMM 指数の 1 ポイント = 2,500 ドル

図 7.2 では図 7.1 の ED 先物と同じ売買 (先物価格も同じ値) を SOFR 先物で行ったとして描いた。以下に記した各項目を 2 つの図で比較することで、SOFR 先物を理解しよう。

利息期間と金利 (右上の破線内)：

- 限月の名称は ED 先物、SOFR 先物共に 12 月限で同じ。
- 両者の**利息期間** (**Accrual period**) は 12 月 20 日から翌年 3 月 20 日までで同じ。(CME ではこの利息期間を **Reference Quarter** "**参照四半期**" と呼ぶ)
- ED 先物の金利は 12 月 18 日 (前決め) に決定されるが、SOFR ターム金利は後決めとなり、3 月 20 日となる。

最終決済価格 発表日：

- ED 先物は Libor 決定日の 12 月 18 日。
- SOFR 先物は利息期間が終了する 3 月 20 日。

最終取引日：

- ED 先物は 12 月 18 日、一方 SOFR 先物は 3 月 19 日。
- SOFR 先物の場合、限月の名称と最終取引日の月が異なる点に注意。
- この違いにより、SOFR 先物の図では 12 月 18 日を " 差金決済 " ではなく " 買い戻し " とした。

9 月 26 日時点の先物の金利：

- ED 先物は 12 月 20 日スタートの 3ヶ月 Libor (いわゆる**フォワード Libor**)。

- SOFR 先物はフォワードスタート (12 月 20 日) の 3ヶ月物 SOFR スワップレート。
- 2 つの先物は利息期間が同じタームレート (Term Rate) を参照していることを理解することが重要。[*5]

これらの比較から 12 月 18 日まで SOFR 先物は ED 先物とおなじ商品性を示していることが判るであろう。繰り返しとなるが、SOFR 先物は SOFR スワップレートと同様 SOFR の “ 前決め ” タームレートとなっている。

7.1.3　3ヶ月 SOFR 先物オプション

3ヶ月 SOFR 先物オプション (CME では **Quaterly Standard Options** と呼ばれ、以下 **SOFR 先物オプション**として参照) の原資産は 3ヶ月 SOFR 先物であるが、このオプションの満期は利息期間 (図 7.2 の右上破線期間、または Reference Quarter) が始まる直前の金曜日であり、2023 年 12 月限 SOFR 先物オプションでは 12 月 15 日 (金曜) であった。

つまり、オプションの満期に利息期間 (SOFR のデイリーコンパウンドする期間) は含まれていない為、前決めのタームレートのオプションとなっている点も含め、ユーロドル先物オプションと同じ商品設計となる。このオプションの具体的な例は 7.2.3 節で紹介する。

7.2　ノーマルモデルでの金利先物オプションの評価

この節では SOFR 先物オプションを計算するモデルとして Bachelier ノーマルモデルを紹介するが、初めに金利の分布について考察する。

7.2.1　ヒストリーデータが示す金利の正規分布性

マイナス金利が発生する以前の 3ヶ月 Ibor は株価や債券価格と同じ様に対数正規分布していることが仮定され、ブラックモデルでオプション価格を算出していた。(式 6.24 参照)

ここで株価を例に対数正規分布の特徴を次のように確認しよう。[*6]

[*5] ここでは “ フォワード ” と “ 先物 ” を同じと考えている。つまり 難解な “ **コンベクシティー調整** ” を無視している。コンベクシティー調整を統一的に理解するには、測度変換を理解した後で Lesniewski [39] 第 4 章 Convexity and CMS が良いだろう。

[*6] 7.2.1 節の記述は Corb [23] の 5.3 節 Questioning Black’s Model for Interest Rate Options を参考にした。

- 株価の変動幅を $dS = S_{t+1} - S_t$ で表すと、株価が高い場合 dS は大きく、株価が低い場合 dS は小さくなる。つまり変動幅の dS は株価の水準に依存している。

- BS モデルでは この依存を排除する為 $dS/S = (S_{t+1} - S_t)/S$ のように株価の収益率をモデル化した。

- 収益率が正規分布する仮定で、元の株価は対数正規分布となった。

要約すると 対数正規分布の特徴は次の 2 点。

(1) 変動幅 dS は株価水準に応じて変動。

(2) 株価収益率は正規分布を仮定。

しかし 3ヶ月 Ibor とは満期 0.25 年の割引債から計算された " 収益率 " である。割引債の価格にブラックモデルを使う場合、上記 (2) より金利は正規分布が仮定される。

上記 (1) に関しては図 7.3 を見てみよう。この図は米ドル 10 年スワップレートを y として、週単位の変動幅 dy を 2 年間 4 つの異なる期間でグラフにしている。(例えば 左上は 1996 年〜1997 年の 2 年間)

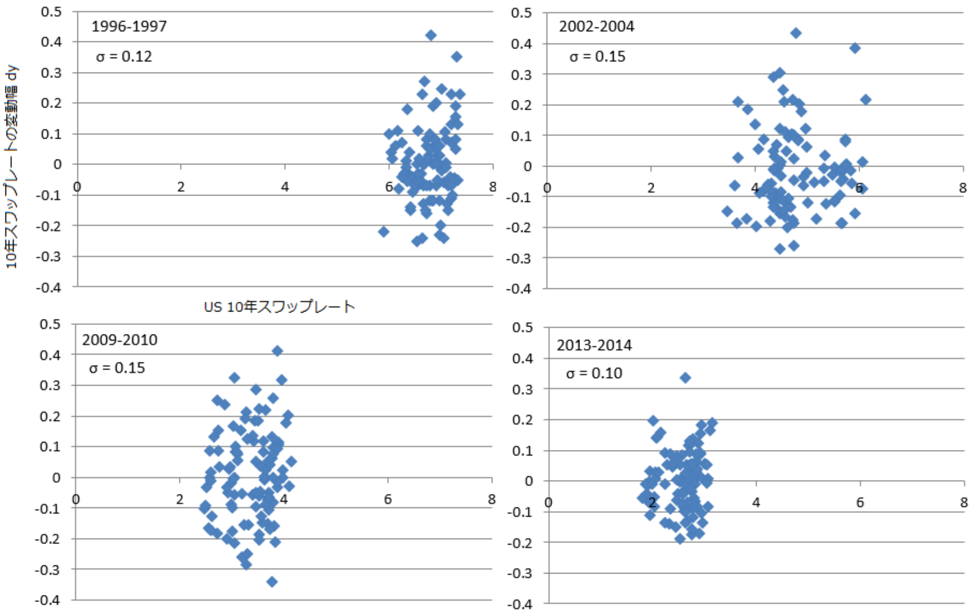


図 7.3: ドル 10 年スワップレート (X 軸) と週単位の変動幅 (Y 軸)
(図中の標準偏差 σ を $\sqrt{52}$ 倍するとヒストリカルベースのノーマルボラティリティとなる)

もし金利が対数正規分布している場合、変動幅 dy が金利水準に応じて、大きくなったり、小さくなったりするはずであるが、この 4 つの図ではどの金利水準でも変動幅 dy の分布にその傾向を読み取ることは出来ないだろう。

このヒストリカルデータも“ 金利の分布は正規分布 ”と仮定される理由の 1 つとなる。

7.2.2　ノーマルモデルとそのオプション価格式

金利 y が正規分布することを金利変化幅 dy を使い、$dy \sim N(\underset{(平均)}{\mu}, \underset{(分散)}{\sigma_y^2})$ で表す。ブラックモデル同様、ノーマルモデルをフォワード金利 (満期 T 年) のモデルとしよう。時点 $t \leq T$ のフォワード金利を F_t で表すと、モデル式は T-forward メジャーで次式となる。(記号を軽くする為、dW_t^T の W の肩の T を省略)

フォワード金利 F_t の正規分布の式

$$
\begin{aligned}
dF_t &\sim N(\underset{(平均)}{0}, \underset{(分散)}{\sigma_N^2 dt}) \quad であり、\\
dF_t &= \underset{(平均)}{0}\, dt + \underset{\substack{(ノーマル\\ボラティリティ)}}{\sigma_N}\, dW_t
\end{aligned}
\tag{7.1}
$$

この方程式の解、つまり“ 時点 T の F_T の値を計算する式 ”を得るには式 (7.1) を次のように時点 0 から T まで積分し、F_T について解けばよい (式 6.62 の展開を参照)。

$$
\begin{aligned}
\int_0^T dF_s &= \int_0^T 0\ ds + \int_0^T \sigma_N\ dW_s \quad より、\\
F_T &= F_0 + \sigma_N\ W_T \\
&= F_0 + \sigma_N \times z\ \sqrt{T}\ , \quad z \sim N(0,1)
\end{aligned}
$$

から、F_T の分布は次式となる。

$$
F_T \sim N(\underset{(平均)}{F_0}, \underset{(分散)}{\sigma_N^2 T}) \tag{7.2}
$$

つまり現在のフォワード金利 F_0 を中心に $\sigma_N\sqrt{T}$ の標準偏差で F_T が正規分布していることを表している。式 (7.1) の**ノーマルボラティリティ**σ_N とは F_t の**年率化**した標準偏差であり、そのことは最後の式で $T = 1$ 年 とすれば $\sigma_N^2 T = \sigma_N^2$ からも確認できる。

式 (7.1) を基にコールオプションのペイオフ $max(F_T - X, 0)$ の期待値を計算させると、次のコール価格式 c(及びプット価格式 p) を導くことができる。(“ コールオプション価格式の導出 ”246 ページを参照)

$$
d = \frac{\overbrace{F_0}^{\substack{フォワード\\金利}} - \overbrace{X}^{\substack{行使\\価格}}}{\underbrace{\sigma_N\sqrt{T}}_{\substack{時点\ T\ の\\標準偏差}}} \tag{7.3}
$$

$$\underset{(\text{コール})}{c} = D_T\Big[\underbrace{(\ F_0\ -X\ )\ \overbrace{N(d)}^{\substack{\text{標準正規}\\\text{分布累積}\\\text{密度関数}}}}_{\text{式 (7.14)}} + \underbrace{\sigma_N\sqrt{T}\overbrace{\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}d^2}}^{\substack{\text{標準正規分布}\\\text{密度関数}}}}_{\text{式 (??)}}\Big] \tag{7.4}$$

$$= D_T\ \underbrace{\sigma_N\sqrt{T}\Big[\ \overbrace{d}^{\substack{\text{満期時}\\\text{デルタ}}}N(d)+n(d)\ \Big]}_{\text{ノーマルコア}} \tag{7.5}$$

$$\underset{(\text{プット})}{p} = D_T\Big[\ -(F_0-X)N(-d)\ +\sigma_N\sqrt{T}\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}d^2}\Big] \tag{7.6}$$

$$= D_T\ \sigma_N\sqrt{T}\Big[\ -dN(-d)+n(d)\ \Big] \tag{7.7}$$

F_0 : フォワード金利や先物価格等　X : 行使価格 (金利)

σ_N : ノーマル ボラティリティ　T : 満期年

D_T : オプション満期日のディスカウントファクター等
(スワップションでは D_T はアニュイティを使用)

標準正規分布 密度関数 : $n(d) = \dfrac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}d^2}$

式 (7.4) 及び (7.6) で $\sigma_N\sqrt{T}$ をカッコの前に出した式が式 (7.5)、(7.7) である。尚、式 (7.5) の D_T を除く部分を**ノーマルコア**として参照する。数式に関して簡単なコメントを記そう。

d : 正規分布の中心であるフォワード金利から行使価格までの差を標準偏差 $\sigma_N\sqrt{T}$ で割ることで算出される d は標準正規分布上の行使価格 X の位置となっている。6.7.1 節で記した標準化の為の変数変換。

- ブラックモデルの d_2 と同じイメージ。

$N(d)$: この項もブラックモデルと同じく、満期時のデルタ。

- ブラックモデルのデルタとノーマルモデルのデルタは異なる。(d は 2 つのモデルで異なる値)

σ_N : 式 (7.5) を使用して計算されるオプション価格が市場建値と同じになるボラティリティであり、インプライド ボラティリティ。

- このボラティリティをノーマルボラティリティと呼び、ブラックボラティリティと区別する。

σ_N のヒストリカルボラティリティの計算 : 式 (6.16) はブラックのボラティリティのヒストリカルボラティリティを算出する式であった。ノーマルボラティリティに対しては次式を使用する。

$$\text{フォワード金利 } F_t \text{の変化幅} : dF_t \quad := \quad F_t - F_{t-1}$$

- ブラックのボラティリティは F_t で割っているので**レシオ ボラティリティ**とも呼ばれる。

(ノーマルモデルのグリークス計算式)

ノーマルモデルのグリークス計算式は以下の通り。

$$\underset{(call)}{\text{デルタ}} : \frac{\partial c}{\partial F} = D_T N(d)\ , \quad put\ :\ -D_T N(-d) \tag{7.8}$$

$$\text{ガンマ} : \frac{\partial^2 c}{\partial F^2} = D_T \frac{n(d)}{\sigma_N \sqrt{T}} \tag{7.9}$$

$$\text{ベガ} : \frac{\partial c}{\partial \sigma_N} = D_T \sqrt{T} n(d) \tag{7.10}$$

$$\text{セータ} : \frac{\partial c}{\partial T} = r_T c - D_T \frac{n(d)\sigma_N}{2\sqrt{T}} \tag{7.11}$$

$$\Big(\ r_T : \text{満期までのリスクフリーレート}\ \Big)$$

注意すべきは左辺分子にある ∂c は受渡代金ではなく、オプション価格 (3ヶ月物 SOFR 金利) の変動幅を示す点である。 (数値例は 7.5.1 節を参照)

(コールオプション価格式の導出 *)

ここでは式 (7.2) からのコールオプションの価格式を導く[*7] 。(記号を軽くする為、分散 $v^2 \equiv \sigma_N^2\ T$ とした)

ペイオフの期待値はブラックモデル (式 6.21) と同様で、

$$\underset{\text{コール}}{\overset{\text{ペイオフ}}{\text{期待値}}} = E\Big[max(F_T - X, 0)\Big] \tag{7.12}$$

$$= \int_X^\infty \underbrace{\frac{1}{\sqrt{2\pi}v}\exp\Big[-\frac{(F_T-F_0)^2}{2v^2}\Big]}_{\substack{\text{正規分布}\\\text{密度関数}}} \times (\underbrace{F_T - X}_{ITM\ \text{の金額}})\ dF_T \tag{7.13}$$

となる。確率変数 F_T を式 (6.49) で標準化させ、

$$x \ \equiv\ \frac{F_T - F_0}{v} \quad \text{より} \quad F_T \ =\ vx + F_0$$

[*7] 式の展開に関して Corb [23] 537 ページを参考にした。

式 (7.13) へ代入する。積分区間の X は $-d$ となり、$dF_T = vdx$ に注意すると、

$$\begin{aligned}
\text{式 (7.13)} \quad \Rightarrow \quad & \int_{-d}^{\infty} \frac{1}{\sqrt{2\pi}v} \exp\Big(-\frac{x^2}{2}\Big) \times (vx + F_0 - X)\ vdx \\
= \quad & v \int_{-d}^{\infty} \frac{1}{\sqrt{2\pi}} x \exp\Big(-\frac{x^2}{2}\Big)\ dx \\
& \qquad + (F_0 - X) \underbrace{\int_{-d}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\Big(-\frac{x^2}{2}\Big)\ dx}_{\text{標準正規分布の累積関数 } N(\cdot)} \\
= \quad & v \int_{-d}^{\infty} \frac{1}{\sqrt{2\pi}} x \exp\Big(-\frac{x^2}{2}\Big)\ dx \\
& \qquad + (F_0 - X) \times \underbrace{N(d)}_{1-N(-d)} \qquad (7.14)
\end{aligned}$$

を得る。上式 1 項目に関しては次のように y を定義し、

$$y \quad \equiv \quad -\frac{x^2}{2} \quad \text{とし} \quad dx \ = \ -\frac{1}{x}dy$$

一時的に x を置換えて、以下のように変形する。

$$\begin{aligned}
\text{式 (7.14) 1 項目} \ \Rightarrow \ & \frac{v}{\sqrt{2\pi}} \int_{-d}^{\infty} x\ \exp(y)\ \Big(-\frac{1}{x}dy\Big) \\
= \quad & \frac{v}{\sqrt{2\pi}} \int_{-d}^{\infty} -\exp(y)\ dy \ = \ \frac{v}{\sqrt{2\pi}} \Big[-\exp(y)\Big]_{-d}^{\infty} \\
= \quad & -\sigma_N \sqrt{T} \Big[\underbrace{\frac{1}{\sqrt{2\pi}} \exp\Big(-\frac{x^2}{2}\Big)}_{\substack{\text{標準正規分布} \\ \text{密度関数 } n(\cdot)}}\Big]_{-d}^{\infty} \ = \ \sigma_N \sqrt{T} \frac{1}{\sqrt{2\pi}} \exp\Big(-\frac{d^2}{2}\Big) \quad (7.15)
\end{aligned}$$

最後の等号は $n(\infty) = 0$ を使用。(この導出に伊藤の公式は使っていない)

7.2.3 3ヶ月 SOFR 先物オプション 数値例

図 7.4 は CME に上場している 3ヶ月 SOFR 先物オプション 12 月限コールを次の条件で評価し、7.2.2 節の計算式からオプション価格 0.1134(B19 セル) を算出した例である。

B2 セル: コール $= 1$, プット $= -1$
B3 セル: 取引日 : 2023 年 9 月 26 日
B4 セル: オプション満期日 : 2023 年 12 月 15 日 (80 日)
C5 セル: 先物価格 : 94.540
C6 セル: 行使価格 : 94.500
B7 セル: ボラティリティ : 0.50%
B8 セル: リスクフリーレート : 5.40% (マネーマーケットレート)

	A	B	C
1	SOFR先物（ノーマルモデル）		（Ｂ列の数式等）
2	pc ： コール=1， プット=-1	-1	
3	取引日	2023/9/26	
4	満期日	2023/12/15	
5	Ｆ：（100-先物価格）/100	5.4600%	価格 94.54000
6	Ｘ：（100-行使価格）/100	5.5000%	価格 94.50000
7	σ：ボラティリティー	0.5000%	
8	ｒ：リスクフリーレート	5.4000%	ﾏﾈｰﾏｰｹｯﾄﾚｰﾄ
9	（基本的計算項目）		
10	満期日数	80	
11	T ： オプション満期年数	0.21918	=B10/365
12	D_T ： ディスカウント	0.988142	=1/(1+B8*B10/360)
13	σ√T	0.23408%	=(B7*B11^0.5)
14	（オプション計算項目）		
15	d*pc	0.17088	=B2*(B5-B6)/B13
16	N(d*pc)	0.56784	=NORMSDIST(B15)
17	n(d*pc)	0.39316	=NORMDIST(B15,0,1,FALSE)
18	ノーマルコア	0.1147%	=B13*(B15*B16+B17)
19	オプション価格	0.1134%	=B12*B18
20	受渡代金（単位100ドル）	283.46	=2500*B19*100
21	デルタ	-56.11%	=B2*B12*B16
22	ガンマ	165.97	=B12*B17/B13
23	ベガ	18.19%	=B12*B11^0.5*B17
24	セータ	-0.20%	=B8*B19-0.5*B7*B17*B12/B11^0.5

図 7.4: 3ヶ月 SOFR 先物オプションの計算例

ただし SOFR 先物も含め、一般の金利先物は IMM 指数で建値されている為、次の 2 点に注意が必要となる。

B5,6：金利の計算： IMM 指数から先物金利 F (B5 セル) と行使金利 X (B6 セル) を次式で算出。

$$\begin{aligned} \text{先物金利 } F \quad &:= \quad 100 - \underset{(IMM\ \text{指数})}{\text{先物価格}} \;=\; 100 - 94.54 \;=\; \underset{(B5\ \text{セル})}{5.460} \\ \text{行使金利 } X \quad &:= \quad 100 - \underset{(IMM\ \text{指数})}{\text{行使価格}} \;=\; 100 - 94.50 \;=\; \underset{(B6\ \text{セル})}{5.500} \end{aligned}$$

B2: ペイオフの修正： この F と X を使うと、金利先物のコールのペイオフは次式のように展開され、金利のプットのペイオフとなる。

$$\begin{aligned} \text{コールのペイオフ} \;&=\; \max\big[\,\text{満期日の先物価格} - \text{行使価格},\ 0\big] \\ &=\; \max\big[\underbrace{(100-F)}_{\text{先物価格}} - \underbrace{(100-X)}_{\text{行使価格}},\ 0\big] \\ &=\; \underbrace{\max\big[\overset{\text{行使金利}}{X} - \overset{\text{先物金利}}{F},\ 0\big]}_{\text{金利のプットのペイオフ}} \end{aligned} \tag{7.16}$$

では図 7.4 の各セルの数式を確認しよう。

A2: pc コール=1, プット=-1： pc は PutCall の略で B2 セルにプット=−1 を設定。

A11: T オプション満期年数： B11 セルの年数はカレンダーベース (Act/365) で算出した年数を使用。

A12: D ディスカウント： B8 セルのリスクフリーレート (Act/360 ベース) はマネーマーケットレートであり、単利でディスカウントファクターを算出。

A15〜A17: d*pc, N(d*pc), n(d*pc)： 式 (7.3) の d に B2 セルの pc を掛けた値を d*pc と表示している。

- これにより、コール、プットを区別する必要なく、N(d*pc), n(d*pc) を算出。
- d の数値例。

$$\underbrace{\overbrace{0.17088}^{d\times pc}}_{(B15)} = \frac{\underbrace{\overbrace{5.4600\%}^{F:\text{ 先物金利}}}_{(B5)} - \underbrace{\overbrace{5.50000\%}^{X:\text{ 行使金利}}}_{(B6)}}{\overset{(B13)}{\underbrace{0.23408\%}_{\sigma_N\sqrt{T}}}} \times \underbrace{\overbrace{-1}^{pc:\text{ コール}}}_{(B2)}$$

A18: ノーマルコア： プット式 (7.7) を用い、ノーマルコア 0.001147 を B18 セルで計算。

$$\underbrace{\overbrace{0.001147}^{\text{ノーマルコア}}}_{(B18)} = \underbrace{\overbrace{0.23408}^{\sigma_N\sqrt{T}}}_{(B13)} \times \left[\underbrace{\overbrace{0.17088}^{-d}}_{(B15)} \times \underbrace{\overbrace{N(0.56784)}^{N(-d)}}_{(B16)} + \underbrace{\overbrace{n(0.39316)}^{n(d)}}_{(B17)} \right]$$

- 計算式からは判りにくいが、この 0.001147 は式 (7.12) のペイオフ $max(X - F_T, 0)$ の期待値から計算された値。
- 時点 T の金利と行使金利 X の差の期待値が 0.001147 となっているので、この数字の単位は F や X の単位と同じ金利となる。

A19, 20: オプション価格, 受渡代金： プット式 (7.7) ではノーマルコアにディスカウントファクターを掛けることで、コール価格 0.001134 を算出。

$$\underbrace{\overbrace{0.001134}^{\text{オプション価格}}}_{(B19)} = \underbrace{\overbrace{0.988142}^{D_T}}_{(B12)} \times \underbrace{\overbrace{0.001147}^{\text{ノーマルコア}}}_{(B18)}$$

- 取引所では額面 100 ドルのオプション価格 0.1134 で建値される。
- オプション受渡代金は脚注*4 で説明した様に額面$10mil ベースで $0.001134 \times 2,500 = 283.46$ ドル。(取引所の建値はこの受渡代金ではない点に注意)

A21〜A24: グリークス： ここで計算されたグリークスの数値例は 7.5.1 節を参照。

7.2.4 QuantLib, scipy.stats によるコード例

QuantLib のバージョン 1.32 ではオーバーナイトレートの SOFR/RFR 用ノーマルモデルのクラスは用意されていない。[*8] ただ計算は非常に単純であり、SciPy の統計学用サブライブラリ **scipy.stats** にある **norm** モジュールを使用したコードを図 7.5 で示しておく。以下簡単にコードを説明しよう。

```
1  from myABBR import *; import myUtil as mu; from scipy.stats import norm
2
3  # 初期値 (pC:call=1 put=-1, futRT:先物金利, stkRT:行使金利)
4  tradeDT,              matDT,            pC, futPRC, stkPRC                  =¥
5  jDT(2023,9,26), jDT(2023,12,15),  -1,   94.54,   94.50; setEvDT(tradeDT)
6
7  # 50bp        5.4%          5.46%              5.5%
8  volRT,       rfRT,         futRT,             stkRT                         =¥
9  50/10000,   5.4/100, (100-futPRC)/100, (100-stkPRC)/100
10
11 # 予備計算: matYR:満期年, SD:Standard Deviation,  matDF:満期日discFactor
12 rfOBJ,_ = mu.ffTSH(tradeDT, rfRT, dcA360, cmpdSPL)
13 matYR   = dcA365.yearFraction(ql.Settings.instance().evaluationDate, matDT)
14 SD      = volRT*np.sqrt(matYR)
15 matDF   = rfOBJ.discount(matDT)
16
17 # オプション価格
18 d1     = pC*(futRT-stkRT)/SD
19 optNPV= matDF*SD*(d1*norm.cdf(d1) + norm.pdf(d1))
20 # グリークス
21 optDL = pC*matDF*norm.cdf(d1)   ; optGM = matDF*norm.pdf(d1)/SD
22 optVG = matDF*np.sqrt(matYR)*norm.pdf(d1)
23 optTH = rfRT*optNPV - 0.5*matDF*norm.pdf(d1)*volRT/np.sqrt(matYR)
24
25 print(f'optNPV:{optNPV:.6%} ', f'受渡金額:{optNPV*100*2500:.2f} ¥n',
26        f'optDL:{optDL:.2%} ' ,    f'optGM:{optGM :.2f} ',
27        f'optVG:{optVG:.2%} ' ,    f'optTH:{optTH:.2%}'             )
```

```
optNPV:0.113385%   受渡金額:283.46
 optDL:-56.11%  optGM:165.97  optVG:18.19%  optTH:-0.20%
```

図 7.5: ノーマルモデル オプション価格式のコード (最も単純なバージョン)

1: from scipy.stats import norm : この import 文を理解する為、モジュール/パッケージに関して、以下を理解しておこう。

import モジュール

- **モジュール**とはメソッド/関数等のコードが書かれている 1 つのファイルのこと。
- "import モジュール" 文でメソッド等が書かれた 1 つのファイルをインポート。

import パッケージ

[*8] IBOR スワップション用の BachelierSwaptionEngine クラス、IBOR cap/floor 用の BachelierCapFloorEngine クラスを前決めタームレートとして使用することは可能。(7.3 節, 7.4 節参照)
尚 QuantLib にクラスが用意されていなくても、クラスを作れば済むことである。(7.5 節参照)

- **パッケージ**とはモジュールがある**ディレクトリ** (またはフォルダー) のこと。(本書ではパッケージを**ライブラリ**として参照。)
- “import パッケージ”文ではパッケージ部分で指定されたディレクトリのファイル全てをインポート。
- サブディレクトリを指定する場合、**import パッケージ. サブパッケージ**のようにドット演算子を使用。

from モジュール/パッケージ import メソッド/モジュール

- まず 包含関係として“ パッケージ ⊇ モジュール ⊇ メソッド/関数等 ”となっていることを理解しよう。
- from の後にはファイル (またはディレクトリ) を指定し、そこに含まれている複数のメソッド (または複数のモジュール) を import。
- この方法で読み込まれたメソッド/モジュールは from 部分のモジュール名/パッケージ名が省略可能。

具体例：

- 19 行目以降に書かれている norm.cdf() 等はモジュール名 scipy.stats を記載せず使用。
- 1 行目の from myABBR import *では myABBR にある全てのメソッドがモジュール名無しで使用可能となる。(この*印は“ 全て ”を表す)

4〜9: 初期値設定： オプション計算に必要な各初期値をもつ変数を設定。(使用した変数名は図 7.4 A2〜B8 セルより類推できるであろう)

12〜15: DF(ディスカウントファクター) 等： 事前準備として 以下 3 項目を算出。

12: rfOBJ, 15: matDF： リスクフリーレート rfRT を使用したフラットイールドカーブのオブジェクト rfOBJ を作成後、15 行目で満期日の DF を matDF へ設定。

- リスクフリーレートは米国のレポレートを想定し、mu.ffTSH 関数の 3, 4 番目の引数を dcA360、cmpdSPL とした。

13: matYR： オプションの残存年数を算出。

- yearFraction の 1 番目の引数 Settings.instance().evaluationDate は 5 行目のコード setEvDT(tradeDT) によって、tradeDT(2023 年 9 月 26 日) である。
- この引数をここでは tradeDT とすれば良いが、図 7.15 のコードの為、敢えて evaluationDate を使用。

14: SD： 図 6.7 で説明した BlackCalculator と同じような計算とする為、ボラティリティ volRT を標準偏差に変換し、SD を設定。(SD は standard deviation の略)

18〜27: オプション NPV, グリークスの計算： 式 (7.3)〜式 (7.11) を使用し、d からセータまでを計算後、プリント文で出力。

7.2.5 ソルバーによるインプライド ボラティリティ算出

ブラック ボラティリティとは異なり、オプション価格から**ノーマル インプライド ボラティリティ**を簡便に計算する解析解が Jäckel [36] で示されている。その解析解は専門知識が必要となる**不完全ガンマ関数** (**Incomplete Gamma function**) の逆関数が使用されている為、本書では説明を省略する。(7.6 節にサンプルコードとその実行例を示したが、実務的に十分使用できる)

一般的にブラック ボラティリティはニュートン法等の数値計算で求めている。ここでは QuantLib のソルバーでノーマル インプライド ボラティリティを見つけるコードを紹介する。QuantLib のソルバーの使い方をこの例で理解しよう。

```
1  # 1. ソルバー用関数の準備 （価格差の算出）
2  def volSLVR(vol) :
3      SD = vol*np.sqrt(matYR) ;   d1 = pC*(futRT-stkRT)/SD
4      calcNPV= matDF*SD*(d1*norm.cdf(d1) + norm.pdf(d1))
5      return tgtNPV-calcNPV
6  # 2. NPV 0.1133%のvol計算
7  tgtNPV = 0.1133/100                # accuracy, guess,  xMin,   xMax
8  impVOL = ql.Brent().solve(volSLVR, 1e-5,     0.001,  5e-5,   0.1)
9                                     # 0.1bp     10bp,   0.5bp,  10%
10 print(f'ImpVOL(tgtNPV={tgtNPV:.5%}): {impVOL:.4%}')
```

```
ImpVOL(tgtNPV=0.11330%): 0.4995%
```

図 7.6: Brent 法によるインプライドボラティリティ計算例

図 7.6 は QuantLib に用意されたソルバーの 1 つ、**Brent**(ブレント) クラス[9] を使用して、インプライド ボラティリティを算出したサンプルコードである。

オプションの条件はこれまでと同じであるが、市場価格が 0.11330%とした場合のインプライド ボラティリティの計算例となり、算出結果は 0.4995%である。初めにこの数値を図 7.5 と比較しよう。(価格を 0.11338%にすれば ボラティリティ0.50%となる)

ではコードを見ていくが、ソルバーの使い方を中心に説明する。まず QLP ドキュメントによれば Brent クラスの説明は次のスクリーンショットとなる。

2〜5: volSLVR(vol) 関数： 8 行目の Brent().solve メソッドの第 1 引数 f を準備。

[9] 方程式の解を求める方法として、データ範囲を半分にし、解がどちら側にあるかを調べることを繰り返すアルゴリズムを**二分法** (**bisection method**) と言う。**ブレント法**はこのような分割法の 1 つで早く安定的に解へ収束するように改良したもの。ブレント法の説明は天谷 [1] 等、最適化のテキストを参照。

- volSLVR 関数は引数 vol (ボラティリティ) からオプション価格 calcNPV を算出し、市場価格 tgtNPV との差を戻す。
- QLP ドキュメントで "single parameter function" とあるように関数 f は引数 1 つとする必要がある。

```
mySolv = Brent()
mySolv.solve(f, accuracy, guess, xMin, xMax)
```

f	single parameter function or function object, the return value is a floating point number
accuracy	Floating-point number representing the solution precision used to stop the calculation
guess	a floating-point number, the initial guess for the root
xMin, xMax	floating point numbers, left and right interval range

7: tgtNPV： 市場価格 0.1133%を tgtNPV 変数に設定。

8〜10: `Brent().solve`(···)： Brent クラスの solve メソッドは 5 つの引数 f、accuracy、guess、xMin、xMax が必要。

- 第 1 引数 f は "function object"。2 行目で定義した関数 volSLVR を指定。
- 第 2 引数 accuracy は **1e-5**$= 1 \times 10^{-5} = 0.1bp$ とし、関数 volSLVR の戻り値が 0.1bp 以下となる vol を解と見做す。(e は exponent の略でこの場合 10 のべき乗を意味)
- 第 3 引数 guess は解の初期値でボラティリティ 10bp を設定。
- 第 4、5 引数は解が存在する範囲を指定。最小 0.5bp (= **5e-5** $= 5 \times 10^{-5}$)、最大 10%を設定。

8, 10: Brent 法による繰り返し： guess で指定されたボラティリティを volSLVR 関数に与え、市場価格 tgtNPV との差が accuracy 以下となるまで Brent 法に従って解の探索を繰り返す。

- accuracy 以下で計算が終わり、そのボラティリティを impVOL 変数に設定。
- guess で指定した値が Brent().solve(···) 経由で関数 f に渡され、計算された値が戻ってくるイメージとなる。

7.3 金利キャップ

この 7.3 節では金利オプションの例として"前決め"の**金利キャップ** (以下**キャップ**) について説明しよう[*10]。金利キャップとは、例えば CME3ヶ月 TermSOFR

[*10] 本書では**前決めの金利キャップ**についてのみ説明し、計算が複雑な**後決めの金利キャップ**は扱わない。どちらがキャップ市場の主力商品となるかは本書執筆時点では不明であった。

(以下 CME を省略。3ヶ月 TermSOFR は 3.4 節を参照) が一定以上の金利になると上回った金利差の分の利益が出るオプションであり、金利のコールオプションのイメージとなる。以下 “上回った金利差” を**インザマネー分の金利**と呼ぶ。

イメージとしては理解しやすいが、詳細となると、複数のオプションを同時に売買する点やオプションの満期とペイオフ発生の時点が異なる点等で、通常のオプション取引よりも若干複雑さを増す。

ただ 金利キャップは変動金利の主役となるであろう 3ヶ月 TermSOFR のオプションであり、アウト オブ ザ マネー (OTM) のボラティリティを観測できるので、金利デリバティブの “基礎素材” として重要な商品である。

7.3.1 キャプレット概要

ここでは満期 1 年、行使価格 5%、想定元本$10mil のキャップの例で計算のイメージを説明しよう。

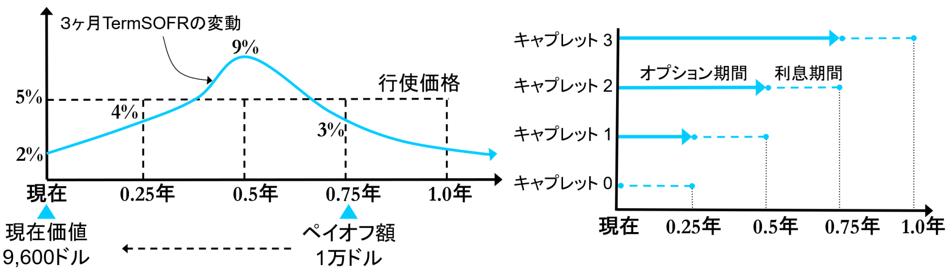


図 7.7: (左) 3ヶ月 TermSOFR の動きとキャップペイオフ額
(右) キャプレットのオプション期間と利息期間

図 7.7 左側では当初 2%の 3ヶ月 TermSOFR が 0.25 年後に 4%、0.5 年後に 9%、0.75 年後には 3%にフィクシング (fixing) される様子を描いている。

このように TermSOFR が推移した場合、キャップの行使価格を上回った期間は 0.5 年から 0.75 年までの 3ヶ月間である。従って、このキャップの投資家は利払日である 0.75 年に

$$\$10mil \times (3\text{ヶ月} \div 360\text{ 日}) \times \underbrace{(9\% - 5\%)}_{\text{インザマネー分の金利}} = \text{約 } 100,000 \text{ ドル}$$

を受け取ることができる。この約 100,000 ドルを現時点まで割り引いた値がキャップ価格となる。例えば、ディスカウントファクター (以下 **DF**) で 0.75 年が 0.96 とすると、キャップの現在価値は約 96,000 ドル (= 100,000 ドル × 0.96)

となる。以上を次式のようにまとめておく。

$$\text{キャップ現在価値} = \overbrace{\underset{(10mil)}{\text{額面}} \times \underset{(0.25\text{ 年})}{\text{期間}} \times \underset{(9\%-5\%)}{\text{インザマネー分の金利}}}^{\text{通常の利息計算と同じ}} \times DF \tag{7.17}$$

ノーマルモデルではオプション満期時の 3ヶ月 TermSOFR が正規分布すると仮定し、この**インザマネー分の金利**をノーマルコアに計算させることになる。

次にこのキャップに含まれるオプションの数と満期を確認しよう。3ヶ月 TermSOFR の利息が計算される期間を**利息期間**（Accrual period）と呼んだが、満期 1 年のキャップは利息期間が 4 つ (= 1 年 ÷ 3 ヶ月) となり、それぞれの利息期間に対応した 4 つのオプション (**キャプレット** Caplet と呼ぶ) で構成される。(図 7.7 右側を参照)

この例で利益が出た 0.5 年〜0.75 年の利息期間に対応するキャプレットは図 7.7 右側のキャプレット 2 番で、オプションの満期は 0.5 年となる。0.5 年時点で 3ヶ月 TermSOFR が 9%に前決めされ、利息期間の金利は確定する。前決め金利なのでペイオフ額は 0.75 年で支払われる。

満期に関して、キャプレット 1 番のオプション満期は 0.25 年となる。キャプレット 0 番は既に満期を迎えたオプションであり、ペイオフ額 (**本源的価値**, Intrinsic value) が払われる 0.25 年後を待っている状態である。

7.3.2　金利キャップのコードと数値例

図 7.8 は取引日 2023 年 9 月 26 日で満期 1 年 5%キャップを Quantlib で評価したコードである。このコードは 3.3.1 節で説明した図 3.13 の 3ヶ月 TermSOFR カーブが構築されている前提となる。初めにコードを概観しよう。

```
 1  # cap条件とSchedule
 2  effDT,            matDT,       capSTK,    ntlAMT,      volRT          = ¥
 3  jDT(2023,9,28),  jDT(2024,9,28),   0.05,   10_000_000,   0.88/100
 4
 5  capSCD  = ql.Schedule(effDT,matDT,pdFreqQ,calUSs,mFLLW,mFLLW,dtGENb,EoMt)
 6  print('Schedule: ', [dd.ISO() for dd in list(capSCD)])        # checking
 7
 8  # Pricing (前提: 3ヶ月TermSOFRカーブ構築済み)
 9  capENG = ql.BachelierCapFloorEngine(sfCrvHDL, mu.sqHDL(volRT))
10  capLeg = ql.IborLeg([ntlAMT], capSCD, TsfIX, dcA360)
11  capOBJ = ql.Cap(capLeg, [capSTK]) ; capOBJ.setPricingEngine(capENG)
12  print(f'capNPV  :    {capOBJ.NPV():,.2f}')
```

```
Schedule:  ['2023-09-28', '2023-12-28', '2024-03-28', '2024-06-28', '2024-09-30']
capNPV  :   42,659.65
```

図 7.8: 3ヶ月 TermSOFR 5% Cap の評価

(前提: 図 3.13 が実行され、3ヶ月 TermSOFR カーブが構築されていること)

2: effDT (起算日)、matDT(満期日)：

- キャップ取引日は SOFR カーブ取引日 (図 3.13 の 2 行目) と同じ 2023 年 9 月 26 日。
- CME TermSOFR は T+2 決済の為 (図 3.13 参照)、キャップの起算日 (effDT) は 2023 年 9 月 28 日、満期日 (matDT) は 1 年後の 2024 年 9 月 28 日を指定。

5, 6: capSCD： effDT、matDT を使用し、Schedule コンストラクタでキャップのキャッシュフロー日を作成。

- EoMt は指標金利 TsfIX(図 3.13 の 16 行目参照) に対応。
- 確認の為、6 行目でキャップのキャッシュフロー日をプリント。

9: BachelierCapFloorEngine： QLP ドキュメント 1 番目のコンストラクタの引数は yieldTermStructure と quoteHandle の 2 つ。(2 番目のコンストラクタに関しては " フラットボラティリティとスポットボラティリティ " で説明)

BachelierCapFloorEngine

ql.BachelierCapFloorEngine(*yieldTermStructure, quoteHandle*)

ql.BachelierCapFloorEngine(*yieldTermStructure, OptionletVolatilityStructure*)

- コンストラクタの 1 つ目の引数 yieldTermStructure にはキャップからのキャッシュフローをディスカウントするカーブを指定する為、OvernightSOFR カーブ sfCrvHDL (図 3.13 の 7 行目で作成) を設定。

    ◇ 1 つ目の引数にディスカウントカーブを指定することはスワップを評価する DiscountingSwapEngine や DiscountingBondEngine に同じ。

- 2 つ目の引数にはボラティリティのハンドル mu.sqHDL(volRT) を指定。
    ◇ 2 つ目の引数としてボラティリティを指定する部分が DiscountingSwapEngine 等と相違。

10: `IborLeg`： 次の QLP ドキュメントの説明の通り、IborLeg は Cap コンストラクタのヘルパークラスで、最低 3 つ nominals, schedule, index の引数が必要。

IborLeg

helper class building a sequence of capped/floored ibor-rate coupon

`ql.IborLeg`(*nominals, schedule, index, paymentDayCounter = DayCounter(), paymentConvention = Following, fixingDays = 0, gearings = 1, spreads, caps, floors, isInArrears, exCouponPeriod, exCouponCalendar, exCouponConvention = Unadjusted, exCouponEndOfMonth = False*)

```
schedule = ql.MakeSchedule(ql.Date(15,6,2020), ql.Date(15,6,2021), ql.Period('6M'))
index = ql.Euribor3M()
leg = ql.IborLeg([100], schedule, index)
```

```
leg = ql.IborLeg([100], schedule, index, ql.Actual360())
```

- このヘルパーを 11 行目の Cap コンストラクタに与えて、Cap オブジェクト作成。
- 3 番目の引数に与えられた TsfIX (図 3.13 の 15 行目で作成) は `IborIndex` コンストラクタで作成されている為、この IborLeg ヘルパーで使用が可能。

11: `Cap`： Cap コンストラクタは floatingLeg, exerciseRates の 2 つの引数が必要。

Cap

`ql.Cap`(*floatingLeg, exerciseRates*)

```
schedule = ql.MakeSchedule(ql.Date(15,6,2020), ql.Date(16,6,2022), ql.Period('6M'))
ibor_leg = ql.IborLeg([100], schedule, ql.Euribor6M())
strike = 0.01
cap = ql.Cap(ibor_leg, [strike])
```

- floatingLeg とは IborLeg で作成した Cap 用ヘルパーを意味。

- 2 番目の引数は債券のクーポン同様、リスト変数として与える。(キャップレートをステップアップやステップダウンさせる為)

11, 12: NPV： 11 行目で作成された capOBJ オブジェクトに 9 行目の capENG をエンジンとしてセットし、12 行目で NPV メソッドにより、キャップ全体の時価 42,659.65 ドルを算出。

(フラットボラティリティとスポットボラティリティ)

BachelierCapFloorEngine コンストラクタ (256 ページ) の 2 番目コンストラクタの 2 番目引数は **OptionletVolatilityStructure** となっている。1 番目コンストラクタとの違いはキャップ市場で次の 2 つのボラティリティが使われることに拠る。

フラットボラティリティ(Flat volatility)： 1 番目のコンストラクタ (引数 quoteHandle) はフラットボラティリティ用。

- フラットボラティリティとはキャップに含まれるすべてのキャプレットを同じボラティリティ(図 7.8 では 88bp) で評価すること。
- 市場の金利キャップ/フロアはフラットボラティリティで建値。
- ただし、フラットボラティリティはキャップの満期年に応じてボラティリティが異なる**ボラティリティのタームストラクチャー**を無視。

スポットボラティリティ(Spot volatility)： 2 番目のコンストラクタ (引数 OptionletVolatilityStructure) はスポットボラティリティ用。

- スポットボラティリティは各キャプレットに異なるボラティリティを与えて評価。
- ボラティリティのタームストラクチャーを考慮した評価法。
- フラットボラティリティで評価したキャップ全体の価値とスポットボラティリティで評価した価値が同じになるように各スポットボラティリティを調整する。(この作業を**ボラティリティのストリッピング**と言う)

キャップに関する説明が長くなる為、スポットボラティリティの数値例は省略する。

7.3.3　キャプレットの詳細

図 7.8 の 11 行目 capOBJ オブジェクトのキャプレット用メソッドを使い、図 7.9 でキャプレットの情報をデータフレーム dfLET で表示させた。

```
1 # capOBJのcaplet用メソッド
2 dfLET = pd.DataFrame( dict(
3             stdDEV = (capOBJ.optionletsStdDev()),        #stdDev=σ√T
4             atmFWD = (capOBJ.optionletsAtmForward()),
5             DF     = (capOBJ.optionletsDiscountFactor()),
6             NPV    = (capOBJ.optionletsPrice())           ))
7 dfLET.style.format(fmtSCF)
```

	stdDEV	atmFWD	DF	NPV
0	0.000000	5.385580%	0.98629210	9,613.00
1	0.004394	5.460013%	0.97285119	12,135.07
2	0.006214	5.257983%	0.95995225	9,762.74
3	0.007625	5.258748%	0.94694952	11,148.83

図 7.9: Cap オブジェクトのキャプレット用メソッド

まず、3 行〜6 行目に書かれた 4 つのメソッドを説明しよう。(2 行目 dict コンストラクタは図 2.5 参照)

3: `optionletsStdDev`：　キャプレット評価に使用した標準偏差。

- ボラティリティは図 7.8 の 2 行目 volRT=0.88%であるが、それに各キャプレットの満期年の平方根を掛けた値を表示。
- 例えば インデックス 3 番の満期年は 0.7507 年 (図 7.10 matYR 列参照) であり、$0.88\% \times \sqrt{0.7507\text{ 年}} = 0.7625\%$。

4: `optionletsAtmForward`：　TermSOFR カーブから算出された各キャプレット Fixing 日のフォワードレート。

5: `optionletsDiscountFactor`：　キャプレットからのキャッシュフローを割り引くディスカウントファクター。

- Overnight SOFR カーブ sfCrvHDL の値であり、Term SOFR カーブの値ではない点に注意。

6: `optionletsPrice`：　各キャプレットの NPV。 この合計額がキャップの NPV。

- “ dfLET.NPV.sum() ” で図 7.8 の capNPV と同じ値。(各自確認)

(Cap ヘルパーのキャスト)

ただ図 7.9 で出力した Caplet 情報は日付関連の情報が不足している為、図 7.10 の 10 行目のように capLeg ヘルパー (図 7.8 の 10 行目で作成) を as_floating_rate_coupon でキャストし追加の情報を出力させた。(このキャストに関しては 2.4.1 節参照)

```
1  # caplet日付関連
2  dfCap = pd.DataFrame({
3      'matYR':       dcA365.yearFraction(tradeDT, cpn.fixingDate()), #maturity year
4      'fixingDate': cpn.fixingDate().ISO(),
5      'accruStart': cpn.accrualStartDate().ISO(),
6      'accruEnd':   cpn.accrualEndDate().ISO(),
7      'payDate':    cpn.date().ISO(),
8      'days':       dcA360.dayCount(cpn.accrualStartDate(), cpn.accrualEndDate()),
9      'TsfDF':      TsfCrvOBJ.discount(cpn.date())            #calc forward rate
10     } for cpn in map(ql.as_floating_rate_coupon, capLeg))
11
12 dfCap = pd.concat([dfCap, dfLET], axis=1)
13 fmtSCF.update(TsfDF='{:.8f}') ; dfCap.style.format(fmtSCF)
```

	matYR	fixingDate	accruStart	accruEnd	payDate	days	TsfDF	stdDEV	atmFWD	DF	NPV
0	0.0000	2023-09-26	2023-09-28	2023-12-28	2023-12-28	91	0.98656929	0.000000	5.385580%	0.98629210	9,613.00
1	0.2493	2023-12-26	2023-12-28	2024-03-28	2024-03-28	91	0.97313833	0.004394	5.460013%	0.97285119	12,135.07
2	0.4986	2024-03-26	2024-03-28	2024-06-28	2024-06-28	92	0.96023558	0.006214	5.257983%	0.95995225	9,762.74
3	0.7507	2024-06-26	2024-06-28	2024-09-30	2024-09-30	94	0.94722901	0.007625	5.258748%	0.94694952	11,148.83

図 7.10: キャプレットの詳細情報

図 7.9 で作成したデータフレーム dfLET は図 7.10 のデータフレーム dfCap に 12 行目で concat させている。

この情報をベースにインデックス 3 番のキャプレットの計算を図 7.11 エクセルで再現させた。各セルの計算を確認しよう。

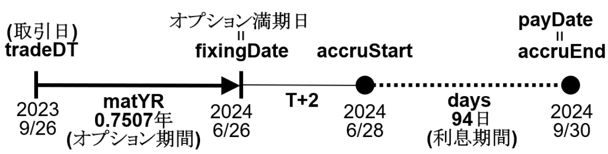
	A	B	C
1	TermSOFR Caplet		（B列の数式等）
2	pc ： コール=1, プット=-1	1	
3	取引日	2023/9/26	2024/1/1
4	満期日 (fixinig日)	2024/6/26	
5	F： (atmFWD)	5.2587%	QLの計算結果(atmFWD列)
6	X：	5.0000%	
7	σ： ボラティリティー	0.8800%	
8	利息期間(日数)	94	QLの計算結果(days列)
9	(基本的計算項目)		
10	満期日数	274	=B4-B3
11	T ： オプション満期年数	0.75068	=B10/365
12	D_T ： ディスカウント	0.94695	QLの計算結果(DF列)
13	$\sigma\sqrt{T}$	0.00762	=B7*B11^0.5 (またはstdDEV列)
14	(オプション計算項目)		
15	d*pc	0.33936	=B2*(B5-B6)/B13
16	N(d*pc)	0.63283	=NORMSDIST(B15)
17	n(d*pc)	0.37662	=NORMDIST(B15,0,1,FALSE)
18	ノーマルコア	0.4509%	=B13*(B15*B16+B17)
19	オプションNPV	11,148.84	=B12*10^7*B8/360*B18

図 7.11: インデックス 3 番のキャプレットの計算例

A2: pc コール=1： キャプレットは金利のコールであり、1 を指定。

A4: 満期日 (fixing 日)、A8: 利息期間、A11: オプション満期年数：

下図はインデックス 3 番キャプレットのタイムラインで図 7.10 の出力から作成。



- キャプレットの満期日は TermSOFR のフィクシング日。
- 日付に関連するセル (A8, A11 等) はこのタイムラインから理解できる為、説明は省略。

A5: F (atmFWD)： 図 3.13 の 3ヶ月 TermSOFR カーブから計算された各利息期間に適用されるフォワードレート。

- 2024 年 6 月 28 日のフォワードレートは式 (2.6) に数値を当てはめ、5.258748%を算出。

$$\underset{\text{フォワードレート}}{\overset{\text{24 年 9 月 30 日償還}}{5.258748\%}} = \frac{1}{94\text{ 日}/360} \times \left(\frac{\overset{\text{スタート}:TsfDF[2]}{0.96023558}}{\underset{\text{満期}:TsfDF[3]}{0.94722901}} - 1 \right)$$

(確認の為に記すが、ノーマルモデルではこのレートが正規分布する)

- フォワードレートを算出するディスカウントファクターは図 7.10 の TsfDF 列の値である点に注意。(3.4.3 節のツーカーブの評価)

A12: D_T ディスカウント： Overnight SOFR カーブ sfCrvHDL から計算された payDate 日のディスカウントファクター。

- キャプレットのペイオフはオプションの満期日ではなく、payDate 列の日付で行われる。

A15: d、A18: ノーマルコア： d とノーマルコアの計算式はこれまでと同様 式 (7.3) と式 (7.5)。

$$\underset{(B15)}{\overset{d}{0.33936}} = \left(\underset{(B5)}{\overset{F:\text{ フォワード}}{5.2587\%}} - \underset{(B6)}{\overset{X:\text{ 行使価格}}{5.0\%}} \right) / \underset{(B13)}{\overset{\sigma_N \times \sqrt{T}}{0.00762}}$$

$$\underset{(B18)}{\overset{\text{ノーマルコア}}{0.4509\%}} = \underset{(B13)}{\overset{\sigma_N \times \sqrt{T}}{0.00762}} \times \left[\underset{(B15)}{\overset{d}{0.33936}} \times \underset{(B16)}{\overset{N(d)}{N(0.33936)}} + \underset{(B17)}{\overset{n(d)}{n(0.33936)}} \right]$$

A19: オプション NPV： キャプレット時価 (NPV) は式 (7.17) と同様、額面と利息期間の日数及び日数計算法から次式となる。

キャプレット時価：$11, 148.84

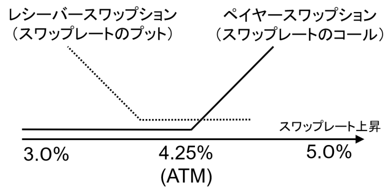
$$= \underset{\substack{\text{支払日}\\\text{ディスカウント}\\\text{ファクター}}}{D_T} \times \underbrace{\text{額面} \times \frac{\text{利息期間の日数}}{360} \times \overset{\text{(インザマネー分の金利)}}{\text{ノーマルコア}}}_{\text{通常の利息計算 (額面×年数×利率)}}$$

$$= \underset{(B12)}{\overset{D_T}{0.94695}} \times \overset{\text{額面}}{\$10mil} \times \underset{(B8)}{\overbrace{94\text{ 日}/360}^{Act/360}} \times \underset{(B18)}{\overset{\text{ノーマルコア}}{0.4509\%}}$$

確認として、以上の説明からキャプレットが TermSOFR のフォワード金利のオプションになっていることが理解できたであろう。7.2 節で説明した SOFR 先物も基本的にフォワード金利を取引している。

従って、**OTC** (Over The Counter, 相対取引) のキャプレットボラティリティは取引所の SOFR 先物コールオプションのボラティリティと ATM (At The Money) で比較可能となる。(OTM ではキャプレットにスキュー/スマイルが発生。スマイルのモデリングは第 8 章を参照)

7.4 スワップション



スワップレートに対するオプションを**スワップション**と言う。スワップレートが高くなるとイン ザ マネーになる固定払いのスワップに対するオプションを**ペイヤースワップション**と呼び、コール式で評価する。反対にスワップレートが低くなるとイン ザ マネーとなる固定受けのスワップに対するオプションを**レシーバースワップション**と呼び、プット式での評価となる。

ノーマルモデルではスワップレートが正規分布すると仮定される。以下ノーマルモデルの数値例を見て行く。(ブラックモデルのスワップション価格式は式 (6.23) を参照)

7.4.1 ノーマルモデルでの数値例

ノーマルモデルのスワップション価格式は式 (7.5)、(7.7) において、D_T 部分をアニュイティに変更した次式となる。

ペイヤースワップション (コール) : c

$$= \text{アニュイティ} \times \underbrace{\overset{\text{原資産の}}{\sigma_N}\sqrt{T}\left[\ dN(d) + n(d)\ \right]}_{\text{ノーマルコア}} \tag{7.18}$$

$$d \ = \ \frac{F_0 - X}{\sigma_N\sqrt{T}} \tag{7.19}$$

F_0 : オプション満期日のフォワードスワップレート

X : 行使価格 (対象スワップのクーポン)

レシーバースワップション (プット) : p

$$= \text{アニュイティ} \times \overset{\text{原資産の}}{\sigma_N}\sqrt{T}\left[\ -dN(-d) + n(d)\ \right] \tag{7.20}$$

式の解釈はノーマルコアで**インザマネー分の金利** **(スワップレート)** を算出し、アニュイティを掛けて 額面 1 に対する価格を求めている。

尚式 (7.18)、式 (7.19) を計算する場合、原資産のアニュイティとフォワードスワップレート F_0 を別途計算する必要がある。

では数値例として 図 7.12 のペイヤースワップションで、原資産は 1 年先スタート、テナー 2 年の SOFR スワップ (指標金利は OvernightSOFR)、その他の条件は下記となる。

B2 セル: コール = 1	
B3 セル: 取引日 :	2023 年 9 月 26 日
B4 セル: 満期日 :	2024 年 9 月 26 日 (366 日)
B5 セル: F: 1y×2y スワップ :	4.2490% (要計算)
B6 セル: X: ATM :	4.2490%
B7 セル: ボラティリティ :	1.3500% (市場建値)
B8 セル: リスクフリーレート :	5.4500% (1 年 SOFRswap レート)
H8 セル: 原資産アニュイティ :	1.80036 (要計算)

- 取引日 2023 年 9 月 26 日の SOFR カーブは図 3.9 で算出したカーブを使用。

	A	B	C	D	E	F	G	H	I
1	SOFRスワップション		(B列の数式等)						
2	pc : コール=1, プット=-1	1			B5: 1年先 2年のスワップレートの計算				
3	取引日	2023/9/26			支払日	δ:テナー		DF	
4	満期日	2024/9/26			(QL算出)	Act/360	F列 計算式	(QL算出)	H列 計算式
5	F: 1yx2yスワップレート	4.2490%	=H9		09/30/24			0.946950	
6	X: ATM	4.2490%	=H9		09/30/25	1.013889	=(A6-A5)/360	0.905250	
7	σ: ボラティリティー	1.3500%			09/30/26	1.013889	=(A7-A6)/360	0.870453	H8=SumProduct
8	r: リスクフリーレート	5.4500%	1ySOFRswap				アニュイティ	1.80036	(F6:F7,H6:H7)
9	(基本的計算項目)						1yx2yスワップ	4.2490%	=(H5-H7)/H8
10	満期日数	366							
11	T : オプション満期年数	1.00274	=B10/365						
12	DT: アニュイティ	1.80036	=H8						
13	σ√T	0.01352	=(B7*B11^0.5)						
14	(オプション計算項目)								
15	d*pc	0.00000	=B2*(B5-B6)/B13						
16	N(d*pc)	0.50000	=NORMSDIST(B15)						
17	n(d*pc)	0.39894	=NORMDIST(B15,0,1,FALSE)						
18	ノーマルコア	0.5393%	=B13*(B15*B16+B17)						
19	オプション価格	0.9710%	=B12*B18						
20	NPV(額面10milドル)	97,095.36	=B19*10^7						
21	デルタ	90.02%	=B2*B12*B16						
22	ガンマ	53.13	=B12*B17/B13						
23	ベガ	71.92%	=B12*B11^0.5*B17						
24	セータ	-0.43%	=B8*B19-0.5*B7*B17*B12/B11^0.5						

図 7.12: SOFR 1y×2y スワップション計算例

(アニュイティとフォワードスワップレートの計算)

初めにアニュイティ (式 2.4) とフォワードスワップレート (式 2.17) を計算する為、

- 前提として、Overnight SOFR カーブの sfCrvOBJ オブジェクトは作成済み。(104 ページ図 3.9 参照)
- 原資産スワップのスケジュールで利払日のリストを作成。
- 利払日を sfCrvOBJ オブジェクトに与え、ディスカウントファクターを算出。

という手順となる。(スワップオブジェクトの作成は不要)

原資産の 2 年スワップの起算日はオプション満期日 (2024 年 9 月 26 日木曜) の 2 営業日後の 2024 年 9 月 30 日、満期日は 2 年応当日の 2026 年 9 月 30 日。この日付を使い、図 7.13 の 1 番セルでスワップ利払日とディスカウントファクターを出力させた。

2,3: fixSCD： Schedule コンストラクタにより、利払日のスケジュールオブジェクト fixSCD を作成。(このコンストラクタは図 3.11 を参照)

```
1  # 原資産スワップ スケジュール
2  swEffDT, swMatDT = jDT(2024,9,30), jDT(2026,9,30)
3  fixSCD = ql.Schedule(swEffDT,swMatDT,pdFreqA,calUSs,mFLLW,mFLLW,dtGENb,EoMf)
4  # for checking
5  scdDTs = [dd.ISO() for dd in fixSCD]              ; print('fixSCD:',scdDTs)
6  DFs    = [sfCrvOBJ.discount(dd) for dd in fixSCD] ; print('   DFs:',nA(DFs))
```

```
fixSCD: ['2024-09-30', '2025-09-30', '2026-09-30']
   DFs: [0.94695 0.90525 0.87045]
```

```
1  # アニュイティ、フォワードスワップレート
2  Annu = mu.calcAnnuity(fixSCD, sfCrvOBJ, dcA360)
3  fwdRT = (DFs[0]-DFs[-1])/Annu
4  print(f'アニュイティ:{Annu:.5f}, フォワードスワップレート:{fwdRT:.5%}')
```

```
アニュイティ:1.80036, フォワードスワップレート:4.24896%
```

図 7.13: (1 番セル) 原資産 1y×2y スワップの日付と DF
(2 番セル) アニュイティとフォワードレートの計算

5,6: 日付リスト scdDTs と DFs： fixSCD オブジェクトを使い、日付リスト scdDTs とディスカウントファクターのリスト DFs を作成。

- これらのデータを図 7.12 E5〜E7 セル (支払日列) と H5〜H7 セル (DF 列) へ転記。

次に図 7.13 の 2 番セルでアニュイティ 1.80036 とフォワードスワップレート 4.2490% を算出した。

2: Annu： アニュイティを算出する関数 **calcAnnuity** は 2.8.2 節において myUtil モジュールへ登録。

3: fwdRT： フォワードスワップレートの計算は式 (2.17) (79 ページ) を参照。

一応、同様の計算を図 7.12 エクセルの H8 セル (アニュイティ)、H9 セル (1Y×2Y スワップレート) で行った。

H8: アニュイティ： H8 セルでテナー (F 列) とディスカウントファクター (H 列) を使用し、エクセル $SumProduct()$ によりアニュイティを計算。

$$\underbrace{1.80036}_{\text{アニュイティ}\ (H8)} = SumProduct(\underbrace{F6:F7}_{\text{テナー}}, \underbrace{H6:H7}_{\substack{\text{ディスカウント}\\\text{ファクター}}})$$

H9: フォワードスワップレート： 1Y×2Y スワップレート 4.2490%を次式で算出。

$$\underbrace{4.2490\%}_{\substack{\text{フォワードスワップ}\\(H9)}} = \frac{\overbrace{0.946950}^{(H5)D_S} - \overbrace{0.870453}^{(H7)D_E}}{\underbrace{1.80036}_{(H8)\ \text{アニュイティ}}}$$

(スワップション価格の計算)

では図 7.12 の A, B 列のスワップション価格の計算を説明しよう。 この A, B 列は次の 2 点を除き、図 7.4 とほぼ同じ計算を行っている。

(1) A5、A6 セル： フォワードレートに H9 セルで計算したフォワードスワップレート 4.2490%を設定。(行使レート X には ATM スワップションを算出する為、同じ値)

(2) A12 セル： A12 セルの D_T は H8 セルで計算したアニュイティ1.80036 を設定。

これらの修正の結果、ペイヤースワップションの計算は次式となる。

A15: d*pc： ATM の場合 ノーマルモデルでは d はゼロ。

$$\underset{(B15)}{\overbrace{0.0}^{d}} = (\underset{(B5)}{\overbrace{4.2490\%}^{F:\ \text{フォワード}}} - \underset{(B6)}{\overbrace{4.2490\%}^{X:\ \text{行使価格}}}) \,/\, \underset{(B13)}{\overbrace{0.01352}^{\sigma_N\times\sqrt{T}}}$$

A18: ノーマルコア、A19: オプション価格、A20: NPV：

- ペイヤースワップション計算式

$$\underset{(B19)}{\overbrace{0.9710\%}^{c:\ \text{コール}}} = \underset{(B12)}{\overbrace{1.80036}^{\text{アニュイティ}}} \times \underbrace{\underset{(B13)}{\overbrace{0.01352}^{\sigma_N\times\sqrt{T}}} \times [\underset{(B15)}{\overbrace{0.0}^{d}} \times \underset{(B16)}{\overbrace{N(0.0)}^{N(d)}} + \underset{(B17)}{\overbrace{n(0.0)}^{n(d)}}]}_{(B18)\ \text{コア}:0.5393\%}$$

```
 1 from myABBR import * ; import myUtil as mu
 2 # 0.SOFRカーブ
 3 crvDATA = [('depo','1d',5.31), ('swap','1m',5.32), ('swap','3m',5.38),
 4            ('swap','6m',5.46), ('swap','1y',5.45), ('swap','2y',5.01),
 5            ('swap','3y',4.67)]
 6 sofrIX, sfCrvOBJ, sfCrvHDL, sfParRATE = mu.makeSofrCurve(crvDATA)
 7 
 8 # 1. 原資産条件、オプション条件、ボラティリティ等
 9 tradeDT,          exprDT,          swEffDT,          swMatDT,      pC       =¥
10 jDT(2023,9,26), jDT(2024,9,26), jDT(2024,9,30), jDT(2026,9,30),    1
11 
12 fwdRT,      spdRT,   payLag,        volHDL,        ntlAMT                   =¥
13 4.249/100,   0.0,       0,    mu.sqHDL(1.35/100), 10_000_000; setEvDT(tradeDT)
14 
15 # 2.原資産オブジェクト
16 fixSCD = ql.Schedule(swEffDT,swMatDT,pdFreqA,calUSs,mFLLW,mFLLW,dtGENb,EoMf)
17 swapOBJ= ql.OvernightIndexedSwap(
18                   pC, ntlAMT, fixSCD, fwdRT, dcA360, sofrIX, spdRT, payLag)
19 # 3. swaption計算
20 swptnOBJ = ql.Swaption(swapOBJ, ql.EuropeanExercise(exprDT))
21 swptnENG = ql.BachelierSwaptionEngine(sfCrvHDL, volHDL)
22 swptnOBJ.setPricingEngine(swptnENG)
23 print(         f'NPV:{swptnOBJ.NPV()           :,.2f}¥n'
24         f'delta(1bp):{swptnOBJ.delta()  /ntlAMT:,.2%}, '
25          f'vega(1bp):{swptnOBJ.vega()   /ntlAMT:,.2%}, '
26            f'annuity:{swptnOBJ.annuity()/ntlAMT:,.5f}'   )
```

```
NPV:97,091.56
delta(1bp):90.02%, vega(1bp):71.92%, annuity:1.80036
```

図 7.14: QuantLib での 1y×2y スワップションの計算

- ノーマルコアの算出値 0.5393%はスワップレートであり、アニュイティを掛けて現在価値に変換。
- 想定元本 10mil ドルを掛け、 NPV=97,095.36 ドル。

A21〜A24: グリークス： グリークスの説明は 7.5.1 節参照。

7.4.2　BachelierSwaptionEngine での計算

図 7.12 エクセルを QuantLib で計算させたコードが図 7.14 となり、計算結果はほぼエクセルの手計算と同一である。ではコードを簡単に見て行こう。

1〜6: `makeSofrCurve`： このコードはこのセルだけで動くようにさせた為、初めに SOFR カーブの構築を行った。

8〜13: 原資産とオプションの条件設定： 図 7.12 エクセルと同じ条件を設定。原資産に関する設定は図 3.4(95 ページ) を参照。以下は注意すべき設定項目。

- exprDT はオプション満期日、swMatDT は原資産スワップの満期日。
- pC に関して、図 3.4 では“ payRcv ”という変数名を使用。

- ATM スワップションの為、fwdRT には正確なフォワードスワップレートを設定。
- payLag=0 を設定。

15〜18: 原資産オブジェクト： payLag=0 のスワップオブジェクトが作成されている点に注意。(その他の引数は図 3.4 を参照)

- このスワップションの計算では、原資産スワップのキャッシュフローをディスカウントする必要は無い為、payLag=0 を設定。(図 3.1 の `paymentLag` の説明参照)

20: `Swaption`： 下図が Swaption コンストラクタの説明となり、必要な引数は swap, exercise の 2 つ。

Swaption

`ql.Swaption`(*swap, exercise, settlementType=ql.Settlement.Physical, settlementMethod=ql.Settlement.PhysicalOTC*)

- 第 1 引数 swap は原資産オブジェクトの swapOBJ を指定。
- 第 2 引数 exercise に関しては、図 6.9 で説明した VanillaOption コンストラクタ参照。

21: `BachelierSwaptionEngine`： 下図が QLP ドキュメントの BachelierSwaptionEngine で、ここでは 1 番目のコンストラクタを使用。

BachelierSwaptionEngine

`ql.BachelierSwaptionEngine`(*yts, quote*)

`ql.BachelierSwaptionEngine`(*yts, swaptionVolatilityStructure*)

- 第 1 引数 yts にはディスカウントカーブの sfCrvHDL を指定。(このケースはシングルカーブである。もしマルチカーブの場合、ディスカウントカーブを指定)
- 第 2 引数 quote にはボラティリティのハンドル volHDL を指定。

23〜26: swptnOBJ のメソッド： 図 7.12 エクセルと関連するメソッドを列挙した。

- アニュイティはあるが、ガンマとセータのメソッドは用意されていない。(アニュイティがあるので、ガンマ, セータ共に手計算は簡単)
- デルタ, ベガ, アニュイティは ntlAMT(想定元本) で割ることで図 7.12 エクセルと同じ単位となる点に注意。